Evaluating Overfit and Underfit in Models of Network Community Structure

Amir Ghasemian, Homa Hosseinmardi and Aaron Clauset

Abstract—A common data mining task on networks is community detection, which seeks an unsupervised decomposition of a network into structural groups based on statistical regularities in the network’s connectivity. Although many methods now exist, the recently proved No Free Lunch theorem for community detection implies that each makes some kind of tradeoff, and no algorithm can be optimal on all inputs. Thus, different algorithms will over- or under-fit on different inputs, finding more, fewer, or just different communities than is optimal, and evaluation methods that use a metadata partition as a “ground truth” will produce misleading conclusions about general accuracy. As a result, little is known about how over- and under-fitting varies by algorithm and input. Here, we present a broad evaluation of over- and under-fitting in community detection, comparing the behavior of 16 state-of-the-art community detection algorithms on a novel and structurally diverse corpus of 406 real-world networks. We find that (i) algorithms vary widely both in the number of communities they find and in their corresponding composition, given the same input, (ii) algorithms can be clustered into distinct high-level groups based on similarities of their outputs on real-world networks, and (iii) these differences induce wide variation in accuracy on link prediction and link description tasks. We then introduce a new diagnostic for evaluating overfitting and underfitting in practice, and use it to roughly divide community detection methods into general and specialized learning algorithms. Across methods and inputs, Bayesian techniques based on the stochastic block model and a minimum description length approach to regularization represent the best general learning approach, but can be outperformed under specific circumstances. These results introduce both a theoretically principled approach to evaluate over- and under-fitting in models of network community structure and a realistic benchmark by which new methods may be evaluated and compared.

Index Terms—Community Detection, Model Selection, Overfitting, Underfitting, Link Prediction, Link Description.

1 INTRODUCTION

Networks are an increasingly important and common kind of data, arising in social, technological, communication, and biological settings. One of the most common data mining tasks in network analysis and modeling is to coarse-grain the network, which is typically called community detection. This task is similar to clustering, in that we seek a lower-dimensional description of a network by identifying statistical regularities or patterns in connections among groups of nodes. Fundamentally, community detection searches for a partition of the nodes that optimizes an objective function of the induced clustering of the network.

Due to broad interest across disciplines in clustering networks, many approaches for community detection now exist [1], [2], [3], and these can be broadly categorized into either probabilistic methods or non-probabilistic methods. Graphical models, like the popular stochastic block model (SBM) [4], typically fall in the former category, while popular methods like modularity maximization [5] fall in the latter. Across these general categories, methods can also be divided into roughly six groups: Bayesian and regularized likelihood approaches [6], [7], [8], spectral and embedding techniques [9], [10], [11], modularity methods [5], information theoretic approaches such as Infomap [12], statistical hypothesis tests [13], and cross-validation methods [14], [15].

Despite great interest, however, there have been relatively few broad comparative studies or systematic evaluations of different methods in practical settings [16], [11], [17] and little is known about the degree to which different methods perform well on different classes of networks in practice. As a result, it is unclear which community detection algorithm should be applied to which kind of data or for which kind of downstream task, or how to decide which results are more or less useful when different algorithms produce different results on the same input [18].

This situation is worsened by two recently proved theorems for community detection [19]. The No Free Lunch (NFL) theorem for community detection implies that no method can be optimal on all inputs, and hence every method must make a tradeoff between better performance on some kinds of inputs for worse performance on others. For example, an algorithm must choose a number of communities \( k \) to describe a given network, and the way it makes this decision embodies an implicit tradeoff that could lead to overfitting on some and underfitting on other networks.

The “no ground truth” theorem states that there is no bijection between network structure and “ground truth” communities, which implies that no algorithm can always recover the correct ground truth on every network [19], even probabilistically. Together, these theorems have broad
implications for measuring the efficacy of community detection algorithms. In the most popular evaluation scheme, a partition defined by node metadata or node labels is treated as if it were “ground truth”, e.g., ethnicity in a high-school social network or cellular function in a protein interaction network, and accuracy on its recovery is compared across algorithms. However, the NFL and “no ground truth” theorems imply that such comparisons are misleading at best, as performance differences are confounded by implicit algorithmic tradeoffs across inputs [19]. Hence, relatively little is known about how over- and under-fitting behavior varies by algorithm and input, past evaluations offer little general guidance, and a new approach to evaluating and comparing community detection algorithms is needed.

Here, we present a broad and comprehensive comparison of the performance of 16 state-of-the-art community detection methods and we evaluate the degree to and circumstances under which they under- or over-fit to network data. We evaluate these methods using a novel corpus of 406 real-world networks from many scientific domains, which constitutes a realistic and structurally diverse benchmark for evaluating and comparing the practical performance of algorithms. We characterize each algorithm’s performance (i) relative to general theoretical constraints, (ii) on the practical task of link prediction (a kind of cross-validation for network data), and (iii) on a new task we call link description. The tradeoff between these two tasks is analogous to the classic bias-variance tradeoff in statistics and machine learning, adapted to a network setting in which pairwise interactions violate independence assumptions.

Our results show that (i) algorithms vary widely both in the number of communities they find and in their corresponding composition, given the same input, (ii) algorithms can be clustered into distinct high-level groups based on similarities of their outputs on real-world networks, and (iii) these differences induce wide variation in accuracy on link prediction and link description tasks. Finally, we introduce and apply a diagnostic that uses the performance on link prediction and link description to evaluate a method’s general tendency to under- or over-fitting in practice.

Our results demonstrate that many methods make uncontrolled tradeoffs that lead to overfitting on real data. Across methods, Bayesian and regularized likelihood methods based on SBM tend to perform best, and a minimum description length (MDL) approach to regularization [20] provides the best general learning algorithm. On some real-world networks and in specific settings, other approaches perform better, which illustrates the NFL’s relevance for community detection in practice. That is, although the SBM with MDL regularization may be a good general algorithm for community detection, specialized algorithms can perform better when applied to their preferred inputs.

2 METHODS AND MATERIALS

Despite well-regarded survey articles [1, 2, 3] there are relatively few comparative studies for model selection techniques in community detection [21, 22, 16, 11, 17, 23] and most of these consider only a small number of methods using synthetic data (also called “planted partitions”) or select only a small number of real-world networks, e.g., the Zachary karate club network, a network of political blogs, a dolphin social network, or the NCAA 2000 schedule network. The narrow scope of such comparative studies has been due in part to the non-trivial nature both of implementing or obtaining working code for competing methods, and of applying them to a large and representative sample of real-world network data sets.

For example, Ref. [11] compared several spectral approaches using planted partition networks and five small well-studied real-world networks. Ref. [17] introduced a generalized message passing approach for modularity maximization [24] that can either use modularity values directly or use leave-one-out cross-validation [15] to infer the number of clusters. Both methods were only evaluated on planted partition models and a small number of real-world networks. Ref. [23] proposed a multi-fold cross-validation technique similar to Refs. [14, 22] and compared results with other cross-validation techniques using synthetic data. Recently, Ref. [25] showed that model selection techniques based on cross-validation are not always consistent with the most parsimonious model and in some cases can lead to overfitting. None of these studies compares methods on a realistically diverse set of networks, or provides general guidance on evaluating over- and under-fitting outcomes in community detection.

In general, community detection algorithms can be categorized into two general settings. The first group encompasses probabilistic models, which use the principled method of statistical inference to find communities. Many of these are variants on the popular stochastic block model (SBM). Under this probabilistic generative model for a graph \( G = (V,E) \) with the size \( N := |V| \), a latent variable denoting the node’s community label \( g_i \in \{1, ..., k\} \), with prior distribution \( q_{a} \) \( (a \in \{1, ..., k\}) \), is assigned to each node \( i \in V \). Each pair of nodes \( i, j \in V \times V \) is connected independently with probability \( p_{g_{i}g_{j}} \). In the sparse case, where \( M := |E| = O(N) \), the resulting network is locally tree-like and the number of edges between groups is Poisson distributed. However, a Poisson degree distribution does not match the heavy-tailed pattern observed in most real-world networks, and hence the standard SBM tends to find partitions that correlate with node degrees. The degree-corrected stochastic block model (DC-SBM) [26] corrects this behavior by introducing a parameter \( \theta \) for each node \( i \) and an identifiability constraint on \( \theta \). In this model, each edge \((i,j)\) exists independently with probability \( p_{g_{i}g_{j}} \theta_{ij} \). The aforementioned planted partition model for synthetic networks can simply be a special case of the SBM with \( k \) communities, when \( p_{g_{i}g_{j}} = p_{in} \) if \( g_{i} = g_{j} \) and \( p_{out} \) if \( g_{i} \neq g_{j} \) [27].

This first group of methods includes a variety of regularization approaches for choosing the number of communities, e.g., those based on penalized likelihood scores [8, 28] various Bayesian techniques including marginalization [29], cross-validation methods with probabilistic models [15], compression approaches like MDL [20], and explicit model comparison such as likelihood ratio tests (LRT) [13].

The second group of algorithms encompasses non-probabilistic score functions. This group is more varied, and contains methods such as modularity maximization and its variants [5, 24], which maximizes the difference between
the observed number of edges within groups and the number expected under a random graph with the same degree sequence; the map equation (Infomap) [12], which uses a two-level compression of the trajectories of random walkers to identify groups; and, various spectral techniques [9], [11], which seek a low-rank approximation of a noisy but roughly block-structured adjacency matrix, among others.

Methods in both groups can differ by whether the number of communities $k$ is chosen explicitly, as a parameter, or implicitly, either by an assumption embedded within the algorithm or by a method of model complexity control. In fact, the distinction between explicit and implicit choices can be subtle as evidenced by a recently discovered equivalence between one form of modularity maximization (traditionally viewed as choosing $k$ implicitly) and one type of SBM (which typically makes an explicit choice) [30]. A brief survey of model selection techniques for community detection is presented in Appendix A.

For the evaluation presented here, we selected a set of 16 representative and state-of-the-art approaches that spans both general groups of algorithms (see Table 1). This set of algorithms is substantially larger and more methodologically diverse than any previous comparative study of community detection methods and covers a broad variety of approaches. To be included, an algorithm must have had reasonably good computational complexity, generally good performance, and an available software implementation.

From information theoretic approaches we selected MDL [20] and Infomap [12]. From the regularized likelihood approaches, we selected the corrected integrated classification likelihood (cICL-HKK) [31]. From among the Bayesian methods Newman and Reinert’s Bayesian (B-NR) [29] and Hayashi, Konishi and Kawamoto’s Bayesian (B-HKK) [31] are selected. From among the modularity based approaches, we selected Newman’s multiresolution modularity (Q-MR) [30], the classic Newman-Girvan modularity (Q) [5], the Zhang-Moore message passing modularity (Q-MP) [24], and the Kawamoto-Kabashima generalized message passing modularity algorithm (Q-GMP) [17]. (We consider a cross-validation technique called leave-one-out (CV-LOO) [15] to be the model selection criterion of Q-GMP.) From the spectral methods, we selected the non-backtracking (S-NB) [9] and Bethe Hessian [10], [11] approaches. For the latter method, we include two versions, S-cBHa and S-cBHm [11], which are corrected versions of the method proposed in Ref. 10. From among the more traditional statistical methods, we selected AMOS [32] and a likelihood ratio test (LRT-WB) [13].

Two classes of algorithms that we do not consider are those that return either hierarchical [31], [34], [35], [50] or mixed-membership communities [37]. Instead, we focus on traditional community detection algorithms, which take a simple graph as input and return a “hard” partitioning of the vertices. As a result, hierarchical decompositions or mixed membership outputs are not directly comparable, without additional assumptions.

As a technical comment, we note that the particular outputs of some algorithms depend on the choice of a prior distribution, as in the Bayesian approaches, or on some details of the implementation. For example, the MDL and Bayesian integrated likelihood methods are mathematically equivalent for the same prior [3], but can produce different outputs with different priors and implementations (see below). However, the qualitative results of our evaluations are not affected by these differences. Finally, we note that the link prediction task is carried out using the learned models themselves, rather than using sampling methods, which improves comparability despite such differences.

To evaluate and compare the behavior of these community detection algorithms in a practical setting, we introduce and exploit the “CommunityFitNet corpus,” a novel data set containing 406 real-world networks drawn from the Index of Complex Networks (ICON) [38]. The CommunityFitNet corpus spans a variety of network sizes and structures, with 30% social, 27% economic, 32% biological, 9% technological, 1% information, and 1% transportation graphs (Fig. 1). Within it, the mean degree of a network is roughly independent of network size, making this a corpus of sparse graphs. In our analysis, we treat each graph as

| Table 1: Abbreviations and descriptions of 16 community detection methods. |
| --- |
| **Abbreviation** | **Ref.** | **Description** |
| Q | [5] | Modularity, Newman-Girvan |
| Q-MR | [30] | Modularity, Newman’s multiresolution |
| Q-MP | [24] | Modularity, message passing |
| Q-GMP | [17] | Modularity, generalized message passing with CV-LOO as model selection |
| B-NR (SBM) | [29] | Bayesian, Newman and Reinert |
| B-NR (DC-SBM) | [29] | Bayesian, Newman and Reinert |
| B-HKK (SBM) | [31] | Bayesian, Hayashi, Konishi and Kawamoto |
| cICL-HKK (SBM) | [31] | Corrected integrated classification likelihood |
| Infomap | [12] | Map equation |
| MDL (SBM) | [20] | Minimum description length |
| MDL (DC-SBM) | [20] | Minimum description length |
| S-NB | [9] | Spectral with non-backtracking matrix |
| S-cBHa | [11] | Spectral with Bethe Hessian, version a |
| S-cBHm | [11] | Spectral with Bethe Hessian, version m |
| AMOS | [32] | Statistical test using spectral clustering |
| LRT-WB (DC-SBM) | [13] | Likelihood ratio test |

1. Available at https://github.com/AGhasemian/CommunityFitNet

Fig. 1. Average degree versus number of nodes for the corpus of 406 real-world networks studied here. Networks were drawn from the Index of Complex Networks (ICON) [38], and include social, biological, economic, technological, and information graphs.
3 NUMBER OF COMMUNITIES IN THEORY AND PRACTICE

3.1 In Theory

A key factor in whether some community detection method is over- or under-fitting to a network is its selection of the number of clusters or communities \( k \) for the network. In the community detection literature, most of the consistency theorems, which provide guarantees on the fraction of mislabeled nodes, apply to dense networks only. For example, Ref. [39], proposes that the fraction of misclassified nodes converges in probability to zero under maximum likelihood fitting, when the number of clusters grows no faster than \( k \) in \( \log N \), with a theoretical prediction of \( \sqrt{N} \).

However, most real-world networks are sparse [40, 41], including all networks in the CommunityFitNet corpus (Fig. 1), meaning results for dense networks are inapplicable. For sparse networks, several lines of mathematical study argue that the maximum number of detectable clusters is \( O(\sqrt{N}) \), explicitly [20, 42, 43], or implicitly as an assumption in a consistency theorem [44, 45, 46].

For example, in the planted \( k \)-partition, the expected number of recoverable clusters grows like \( O(\sqrt{N}) \) [44, 45]. (For convex optimization on the planted \( k \)-partition model, a tighter \( O(\log N) \) bound on the number of clusters has also been claimed [37], although this result is not rigorous.) This theoretical limit is remarkably similar to the well-known resolution limit in modularity [48], which shows that modularity maximization will fail to find communities with sizes smaller than \( O(\sqrt{M}) \). Hence, the expected number of modularity communities in a sparse graph is also \( O(\sqrt{M}) \). An argument from compression leads to a similar bound on \( k \) [20, 42]. Specifically, the model complexity of a stochastic block model is \( \Theta(k^2) \), which under a minimum description length analysis predicts that \( k = \Theta(\sqrt{M}) \). This statement can also be generalized to regularized likelihood and Bayesian methods. Although none of these analyses is fully rigorous, they do point to a similar theoretical prediction: the number of recoverable communities in a real sparse network should grow like \( O(\sqrt{N}) \). Different algorithms, of course, may have different constants of proportionality or different sub-asymptotic behaviors. In our evaluations, we use a constant of 1 as a common reference point.

As a technical comment, we note that the maximum number of clusters found is not the same as the number of identifiable clusters under the information-theoretic limit to detectability [49]. For example, as a result of a resolution limit, an algorithm might merge two clusters, but still infer the remaining clusters correctly. In other words, the network’s communities exist in the detectable regime but the output has lost some resolution.

3.2 In Practice

We applied our set of 16 community detection methods to the 406 real-world networks in the CommunityFitNet corpus. For methods with free parameters, values were set as suggested in their source papers. We then binned networks by their size \( N \) or \( M \) and for each method plotted the average (Fig. 2) and maximum number (Fig. 3) of inferred communities as a function of the number of nodes \( N \) and edges \( M \).

In both figures, solid lines show the theoretically predicted trends of \( \sqrt{N} \) and \( \sqrt{M} \). Two immediate conclusions are that (i) the actual behavior of different algorithms on the same input is highly variable, often growing non-monotonically with network size and with some methods finding 10 times as many communities as others, but (ii) overall, the number of communities found does seem to grow with the number of edges, and perhaps even roughly like the \( \sqrt{M} \) pattern predicted by different theoretical analyses (Section 3.1). Furthermore, the empirical trends are somewhat more clean when we consider the number of communities \( k \) versus the number of edges \( M \) (Figs. 2b and 3b), suggesting that the mean degree of a network impacts the behavior of most algorithms.
From this perspective, algorithms visually cluster into two groups, one of which finds roughly 2-3 times as many communities as the other. The former group contains the methods of Q, Q-MR, Infomap, LRT-WB and AMOS, all of which find substantially more clusters than methods in the latter group, which includes B-NR, B-HKK, cICL-HKK, MDL, spectral methods, Q-MP and Q-GMP. In fact, first group of methods often return many more clusters than we would expect theoretically, which suggests the possibility of consistent overfitting. As a small aside, we note that the expected number of communities found by Q-MR and Q are different, because they are known to have different resolution limits [50]. More generally, the aforementioned groups, and their tendency to find greater or fewer communities aligns with our dichotomy of non-probabilistic (more communities) versus probabilistic (fewer communities) approaches. Additionally, we note that the AMOS method failed to converge on just over half of the networks in the CommunityFitNet corpus, returning an arbitrary maximum value instead of $k$. Because of this behavior, we excluded AMOS from all our subsequent analyses.

Figure 3 shows more clearly that the maximum number of clusters detected by Q-MR and Infomap are nearly identical. Furthermore, these methods find the same average number of clusters over more than 400 networks (Fig. 2). This behavior has not previously been noted, and suggests that Q-MR and Infomap may have the same or very similar resolution limits.

In general, algorithms with similar formulations or that are based on similar approaches show similar behavior in how $k$ varies with $M$. For instance, beyond Q-MR and Infomap’s similarity, we also find that MDL, various regularized likelihood methods, and the Bayesian approaches find similar numbers of communities. Spectral methods appear to behave similarly, on average (Fig. 3), to the Bayesian approaches. However, spectral approaches do seem to overfit for large network sizes, by finding a maximum number of communities that exceeds theoretical predictions, in contrast to the Bayesian approaches.

Looking more closely at similar methods, we observe small differences in the number of clusters $k$ they return as a function of network size $N$, which must be related to differences in their implicit assumptions. For example, B-HKK, B-NR and cICL-HKK often agree on the number of communities for networks with a smaller number of edges, but they disagree for networks with a larger number of edges (Fig. 2). Due to a more exact Laplace approximation with higher order terms, B-HKK penalizes the number of clusters more than B-NR and cICL-HKK, which limits the model space of B-HKK to smaller models that correspond to fewer communities. This tradeoff is a natural one, as approaches that penalize a model for greater complexity, like in B-HKK, are intended to reduce the likelihood of overfitting, which can in turn increase the likelihood of underfitting.

Returning to the Q-MR method, we inspect its results more carefully to gain some insight into whether it is overfitting or not. Ref. [30] proves that Q-MR is mathematically equivalent to a DC-SBM method on a $k$-planted partition space of models. The Q-MR algorithm works implicitly like a likelihood-maximization algorithm, except that it chooses its resolution parameter, which sets $k$, by iterating between the Q and DC-SBM formulations of the model. Evidently, this approach does not limit model complexity as much as a regularized likelihood and tends to settle on a resolution parameter that produces a very large number of communities. This behavior highlights the difficulty of characterizing the underlying tradeoffs that drive over- or under-fitting in non-probabilistic methods. We leave a thorough exploration of such questions for future work.

The variation across these 16 methods of the average and maximum number of communities found, provides suggestive evidence that some methods are more prone to over- or under-fitting than others, in practice. The broad variability of detected communities by different methods applied to the same input is troubling, as there is no accepted procedure for deciding which output is more or less useful.
4 QUANTIFYING ALGORITHM SIMILARITY

Although algorithms can be divided into groups based on their general approach, e.g., probabilistic and non-probabilistic methods (see Section 1), such conceptual divisions may not reflect practical differences when applied to real data. Instead, a data-driven clustering of algorithms can be obtained by comparing the inferred labels of different methods applied to a large and consistent set of real-world networks. That is, we can use our structurally diverse corpus to empirically identify groups of algorithms that produce similar kinds of community structure across different data sets. We quantify algorithm similarity by computing the mean adjusted mutual information (AMI) [51] between each pair of methods for the communities they recover on each network in the CommunityFitNet corpus. We then apply a standard hierarchical clustering algorithm to the resulting matrix of pairwise similarities in algorithm output (Fig. 4a). Using the unadjusted or normalized mutual information (NMI) yields precisely the same clustering results, indicating that these results are not driven by differences in the sizes of the inferred communities, which are broadly distributed (Fig. 4b).

The derived clustering of algorithms shows that there is, indeed, a major practical difference in the composition of communities found by probabilistic versus non-probabilistic methods. In fact, methods based on probabilistic models typically find communities that are more similar to those produced by other probabilistic methods, than to those of any non-probabilistic method, and vice versa. This high-level dichotomy indicates a fairly strong division in the underlying assumptions of these two classes of algorithms.

The non-probabilistic methods group further subdivides into subgroups of spectral algorithms (S-cBHm, S-NB, and S-cBHm), consensus-based modularity algorithms (Q-MP and Q-GMP), traditional statistical methods (AMOS and LRT-WB), and finally other non-probabilistic methods (including Infomap, Q, and Q-MR). The fact that algorithms themselves cluster together in the kind of outputs they produce has substantial practical implications for any application that depends on the particular composition of a network clustering. It also highlights the subtle impact that different classes of underlying assumptions can ultimately have on the behavior of these algorithms when applied to real-world data.

5 EVALUATING COMMUNITY STRUCTURE QUALITY

To evaluate and compare the quality of the inferred clusters for a particular network, we need a task that depends only on a network’s connectivity and that can reveal when a method is over- or under-fitting these data. (Recall that the NFL and “no ground truth” theorems of Ref. [19] imply that a comparison based on node metadata cannot be reliable.) For relational data, a common approach uses a kind of network cross-validation technique, called link prediction [52], in which some fraction of the observed edges in a network are “held out” during the model-fitting stage, and their likelihood estimated under the fitted model.

We note, however, that there is as yet neither a consensus about how to design such a task optimally nor a theoretical understanding of its relationship to model fit. For example, it was recently shown that selecting the most parsimonious
model in community detection, by maximizing model posterior probability, can correlate with selecting the model with highest link prediction accuracy \cite{25}. The same results show that under an adversarial setting, optimizing on link prediction accuracy can lead to overfitting. Furthermore, the theoretical implications are unknown for distinct approaches to construct a held-out data set from a single network, for example, holding out a uniformly random subset of edges or all edges attached to a uniformly random subset of nodes. Although there are strong theoretical results for cross-validation and model selection for non-relational data, whether these results extend to networks is unclear as relational data may violate standard independence assumptions. Theoretical progress on this subject would be a valuable direction of future research.

Here, we introduce an evaluation scheme based on a pairing of two complementary network learning tasks: a link prediction task, described above and in Box 1 and a new task we call link description, described below and in Box 2. The goal of these tasks is to characterize the behavior of methods in general, i.e., a method’s general tendency to over- or under-fit across many real networks, rather than to evaluate the quality of a fit to any particular network. A key feature of this scheme is that a method cannot be perfect at both tasks, and each method’s tradeoff in performance across them creates a diagnostic to evaluate the method’s tendency to over- or under-fit real data.

In our evaluation, each method uses a score function to estimate the likelihood that a particular pair of nodes \( ij \) should be connected. Most algorithms optimize some underlying objective function in order to sort among different community partitions. In our main evaluation, we use model-specific score functions, which are based on the method’s own objective function. This choice ensures that each method makes estimates that are aligned with its underlying assumptions about what makes good communities.

We then compare these model-specific results with those derived from a SBM-based scoring function. This comparison to a fixed reference point allows us to better distinguish between poor generalizability being caused by a low-quality model of the network’s structure and the selection of a low-quality partition or set of communities.

### 5.1 Model-specific Link Prediction and Description

#### Link prediction

When a graph \( G = (V, E) \) has been sampled to produce some \( G' = (V, E') \), where \( E' \subset E \), the goal of link prediction is to accurately distinguish missing links (true positives) from non-edges (true negatives) within the set of unobserved connections \( ij \in V \times V \setminus E' \). Link prediction is thus a binary classification task and its general accuracy can be quantified by the area under the ROC curve (AUC).

For our evaluation, we parameterize this classification accuracy by \( \alpha \in (0, 1) \), which determines the fraction of edges “observed” (equivalently, the density of sampled edges) in \( G' = (V, E') \), where \(|E'| = \alpha|E|\) is a uniformly random subset of edges in the original graph \( G \). For a given method \( f \), its AUC as a function of \( \alpha \) which we call an “accuracy curve,” shows how \( f \) performs across a wide variety of such sampled graphs, ranging from when very few edges are observed (\( \alpha \to 0 \)) to when only a few edges are missing (\( \alpha \to 1 \)).

Each network \( G \) in our corpus produces one such accuracy curve, and we obtain a single “benchmark performance curve” for each method by computing the mean AUC at each value of \( \alpha \) across curves produced by the networks in the CommunityFitNet corpus. When computing the AUC, we break ties in the scoring function uniformly at random.

Box 1 describes the link prediction task in detail.

In this setting, the AUC is preferred over precision-recall because we are interested in the general performance of these classifiers, rather than their performance under any specific prediction setting. Evaluating other measures of accuracy is beyond the scope of this study. Comparing benchmark performance curves across community detection methods quantifies their relative generalizability and predictiveness, and allows us to assess the quality of choice each method makes for the number of clusters it found in Section 3.2.

In our evaluation here, we exclude S-cBHa and S-cBHm because they produce very similar results to the S-NB method, LRT-WB because of its high computational complexity, and AMOS and Q-GMP because of convergence issues. All other methods are included.

We now define a model-specific score function for each method. Each score function uses the corresponding method \( f \) to define a model-specific function \( s_{ij} \) that estimates the likelihood that a pair of nodes \( ij \) should be connected.

For regularized likelihood/Bayesian approaches of the SBM (cICL-HKK, B-NR and B-HKK), we follow the same scoring rule as in Ref. \cite{53}. Specifically, the score \( s_{ij} \) assigned to an unobserved edge between nodes \( i \) and \( j \), for \( (i,j) \notin E' \) with community assignments of \( g_i \) and \( g_j \), respectively, is given by \[ s_{ij} = \frac{\ell_{g_i,g_j} + 1}{r_{g_i,g_j} + 2}, \] where \( \ell_{g_i,g_j} \) is the number of edges in the observed network \( G' \) between the groups \( g_i \) and \( g_j \) and \( r_{g_i,g_j} \) is the maximum possible number of links between the groups \( g_i \) and \( g_j \).

For DC-SBM we define \( s_{ij} = \theta_i \ell_{g_i,g_j} \), where \( \theta_i \) is the normalized degree of node \( i \) with respect to total degree of its type as the maximum likelihood esti-
mation of this parameter. For all the Q methods, Infomap, and MDL we define the scores as the contribution that the added unobserved edge would make to their corresponding partition score functions. For the Q methods, we compute the increase in the modularity score due to the added unobserved edge, while for Infomap and MDL we compute the decrease of their objective functions. For each of these methods, the contribution of each pair of nodes is computed under the partition obtained prior to adding the candidate pair as an edge.

There is no commonly accepted link prediction approach for spectral clustering algorithms that is independent of metadata. Although there are some non-linear embedding methods for link prediction like node2vec [54], here we focus on linear decomposition techniques. For spectral clustering, we introduce and use a new link prediction technique based on eigenvalue decomposition. Let the adjacency matrix of the observed graph \( G' \) be denoted by \( A' \). This matrix can be decomposed as \( A' = V \Lambda V^T \), where \( V = [v_1, v_2, ..., v_N] \) with \( v_i \) as \( i \)th eigenvector of matrix \( A' \) and matrix \( \Lambda = \text{diag}[\lambda_1, \lambda_2, ..., \lambda_N] \) is the diagonal matrix of eigenvalues of \( A' \), where \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \).

To define a new scoring function, we use a low-rank matrix approximation of \( A' \) using the \( k \) largest eigenvalues and their corresponding eigenvectors, i.e., we let \( \tilde{A}' = [v_1, v_2, ..., v_k] \text{diag}[\lambda_1, \lambda_2, ..., \lambda_k] [v_1, v_2, ..., v_k]^T \), where \( k \) can be inferred using a model selection spectral algorithm. The spectral method scoring rule \( s_{ij} \) assigned to an unobserved edge between nodes \( i \) and \( j \), for \( (i,j) \notin E' \), is the corresponding entry value in low-rank approximation.

The performance of each method is assessed by numerically computing its benchmark performance curve, using the 406 real-world networks in the CommunityFitNet corpus. Exactly calculating a single accuracy curve for a sparse graph \( G \) takes \( \Omega(N^2) \) time, which is prohibitive for large networks. However, each AUC in a curve may be accurately estimated using Monte Carlo, because the AUC is mathematically equivalent to the probability that a uniformly random true positive is assigned a higher score than a uniformly random true negative. In all of our experiments, an accuracy of \( \pm 0.01 \) is sufficient to distinguish performance curves, requiring 10,000 Monte Carlo samples.

Community detection methods that are prone to overfitting (underfitting) will tend to find more (fewer) communities in a network than is optimal. Hence, the induced partition of the adjacency matrix into within- and between-group blocks will over- (under-) or under- (under-) estimate the optimal block densities. This behavior will tend to produce lower AUC scores for the prediction of uniformly held-out pairs in the corresponding evaluation set. That is, a lower benchmark performance curve indicates a greater general tendency to over- or under-fit on real-world networks.

**Link description**

The goal of link description is to accurately distinguish observed edges \( E' \) (true positives) and observed non-edges \( V \times V \setminus E' \) (true negatives) within the set of all pairs \( ij \in V \times V \). That is, link description asks how well a method learns an observed network, and it is also a binary classification task.
tend to over-partition the network, finding many more communities than is optimal.

We also find that Q and Q-MR exhibit nearly identical benchmark performance curves for link prediction. Although Q has a larger resolution limit, which leads to fewer inferred clusters, our results suggest that Q still finds more communities than is optimal, especially compared to other methods. Evidently, these two methods tend to misinterpret noisy connectivity within communities as indicating the presence of substructure deserving of additional fitting, and they do this in a similar way, leading to similar numbers of communities and very similar link prediction curves. This behavior may reflect the common assumptions of these methods that communities are assortative (edges mainly within groups) and that the between-group edge densities are uniform. If this assumption is not reflected in the input network’s actual large-scale structure, these methods can overfit the number of communities (Figs. 2 and 3), subdividing larger groups to find a partition with the preferred structure.

The best benchmark performance curves for link prediction are produced by Bayesian methods (B-NR (SBM), B-HKK, and cICL-HKK) and MDL methods (both DC-SBM and SBM). And the SBM methods generally outperform the DC-SBM methods, except for the DC-SBM with MDL regularization, which yields the best overall benchmark curve for nearly every value of $\alpha$.

Such a difference is surprising since the number of inferred clusters for both Bayesian and regularized-likelihood methods is nearly identical (Fig. 2b), and the precise composition of the clusters is very similar (Fig. 4). However, these methods use different score functions to estimate the likelihood of a missing edge, and evidently, those based on more general rules perform better at link prediction. For instance, B-NR (SBM) assigns the same scores to the links inside each cluster, whereas B-NR (DC-SBM) assigns higher scores to the links connected to high degree nodes. In B-NR, the emphasis on modeling node degrees by the DC-SBM score function leads to worse performance. In contrast, the MDL technique, while based on the same underlying probabilistic network model, assigns higher scores to edges that produce a better compression of the input data (shorter description length). Hence, the MDL score function prefers adjacencies that decrease the model entropy without increasing model complexity, meaning that it predicts missing links in places with better community structure. The MDL approach to controlling model complexity, particularly in the DC-SBM score function, is more restrictive than most Bayesian approaches, but it evidently leads to more accurate link prediction (Fig. 5).

The benchmark performance curves for Infomap, spectral clustering (S-NB), and B-NR (DC-SBM) are similar, especially for modest or larger values of $\alpha$, and are close to the middle of the range across algorithms. Furthermore, we find that the curves of B-HKK (SBM) and cICL-HKK (SBM) are similar to, but lower than B-NR (SBM). There are two possibilities for this behavior: (i) the number of inferred clusters is inaccurate, or (ii) these methods perform poorly at link prediction. Because the score functions of B-HKK (SBM), cICL-HKK (SBM), and B-NR (SBM) are similar, the lower link prediction benchmark performance is more likely caused by a low-quality set of inferred clusters, due to over- or under-fitting.

The benchmark results for link description (Fig. 5b) show that B-HKK (SBM) and cICL-HKK (SBM) perform relatively poorly compared to most other methods, which suggests that they must tend to under-fit on networks. This behavior is likely driven by their larger penalty terms, e.g., compared to methods like B-NR (SBM), which will favor finding a smaller number of clusters in practice (Fig. 2). This behavior will tend to aid in link prediction at the expense of link description. We note that Ref. [31] introduced a better approximation for B-HKK (SBM)'s penalty terms, which might suggest that the method would find more optimal partitions in theory. However, our results show that this is not the case in practice, and instead this method illustrates a tradeoff in which a greater penalty for model complexity, by over-shrinking or over-smoothing the model space, can lead to poor performance in practical settings.

The best benchmark performance for link description is given by Infomap first, followed by the Bayesian tech-
Fig. 6. Benchmark performance curves using a SBM-based score function for (a) link prediction and (b) link description tasks. Each curve shows the mean AUC for a different community detection method across 406 real-world networks for a given fraction $\alpha$ of observed edges in a network.

The relative performance of Q, Q-MR, and Infomap versus other methods on these tasks provides an opportunity to understand how an algorithm’s assumptions can drive over- and under-fitting in practice. By definition, the partitions found by Infomap and modularity-based methods like Q and Q-MR will tend to have highly assortative communities and a uniformly low density of edges between communities. Such a partition must perform poorly at modeling the few edges between these clusters; hence, as the density of these edges increases with $\alpha$, these methods’ link description performance must tend to decrease. In contrast, nearly all other methods generally perform better at link description as more edges are sampled, except for B-NR (DC-SBM), whose performance is relatively independent of $\alpha$. As a group, the probabilistic methods have the flexibility to model different rates of between-group connectivity, but this behavior requires sufficient observed connectivity in order to estimate the corresponding densities. As a result, these models are less data efficient than are modularity and spectral methods at describing the observed structure, especially in the sparsely-sampled regime (low $\alpha$).

The exception to this interpretation is Q-MP, which exhibits a poor performance on both tasks (Fig. 5a,b). These tendencies can be understood in light of the relatively small number of communities Q-MP tends to find (Figs. 2 and 3), which suggests that it tends to substantially under-fit to the data. In fact, Q-MP uses a consensus of many high-modularity partitions in order to explicitly avoid overfitting to spurious communities. Evidently, this strategy tends to find far fewer communities than is optimal for either link description or link prediction. The Q-GMP method may perform better, as it controls the bias-variance tradeoff through a learning phase on its parameters. Due to poor convergence behavior with this method, however, we leave this line of investigation for future work.

5.2 Performance Under a Common Score Function

Comparing link prediction and link description benchmark performance curves of 11 state-of-the-art community detection methods reveals substantial evidence that most methods tend to over- or under-fit to networks, to some degree. However, poor performance at either task could also be the result of a poor pairing of a particular score function with the particular communities an algorithm finds.

A valuable check on our above results is to test the performance of the identified communities under a common score function. This experiment thus serves to remove differences in the way the various scoring functions utilize the same partition structure. Specifically, we repeat both link prediction and description evaluation tasks, using the community partitions identified by each of the 11 algorithms for each network in the corpus, and then applying the SBM score function from Section 5.1 to construct the benchmark performance curves. Although any score function could be used as such a reference point, the SBM score function has the attractive property that it yielded high general performance in previous section for link prediction.

Results

The relative ordering of the benchmark performance curves under the common score function for the link prediction and description evaluations (Fig. 6) differs in interesting ways from that of the model-specific evaluation (Fig. 5). We note that the performance curves for the SBM-based methods are unchanged as their score function is the same in both settings.
In link prediction, the previous performance gap between the B-NR (DC-SBM) and B-NR (SBM) methods is much smaller, and B-NR is now the best overall method by a sizable margin. The MDL (DC-SBM) method, which produced the best model-specific results for link prediction, performs substantially worse under the SBM-based score function on both tasks. Of course, SBM-based methods should produce communities that exhibit better performance under an SBM-based score function than would other methods. But the DC-SBM in particular was originally designed to find more reasonable communities than the SBM, by preventing the model from fitting mainly to the network’s degree structure [26]. The worse performance by the DC-SBM communities on link prediction in this setting indicates these methods’ allowance of a lower entropy in the inferred block structure acts to over-regularize the communities from the perspective of the SBM. Furthermore, unlike the SBM score function, the MDL (DC-SBM) score function (used in Fig. 5) depends on the model complexity, the inclusion of which evidently serves to improve link predictions at all values of α. However, link prediction using the inferred communities alone appears to be a slightly unfair evaluation of the DC-SBM (also suggested by Ref. [55]).

Turning to other methods, recall that Infomap and Q-MR found similar numbers of communities and had similar accuracies in the model-specific link prediction task (Fig. 5). Under the common SBM-based score function, we find that Infomap, Q-MR, and Q exhibit nearly identical performance on both link prediction and description tasks. In light of our previous discussion of the tendency of modularity-based methods to overfit, this similarity, which must derive from these methods all identifying similar community structures in networks, provides additional evidence that all three methods tend to overfit real data.

Finally, the S-NB method shows unusual behavior: in link prediction, its performance is similar under both score functions; and, in link description, its performance under its own model-specific score function is replaced with a non-monotonic performance curve, which is better at lower values of α than at higher values. The behavior at smaller values of α, when the sampled networks are relatively more sparse, is consistent with a tendency for S-NB to under-fit in this regime, in agreement with past results that suggest that spectral methods tend to under-fit when communities are unbalanced [11]. However, the change at larger values of α indicates that, as more edges are sampled, this spectral technique qualitatively changes in its behavior. Recall that the maximum number of clusters inferred by spectral methods for large networks exceeds the theoretical bound (Fig. 3), which indicates a tendency to overfit. Hence, the relatively worse performance at larger values of α on both tasks suggests that spectral techniques behave differently across different settings, overfitting in large sparse networks, underfitting when communities are unbalanced, and “well-fitting” when communities are balanced. An algorithm that exhibits this kind of context-dependent behavior is deemed to exhibit an “uneven” fit.

| Box 3: Behavior of Community Detection methods |
|-----------------------------------------------|
| Well-fitted | link prediction: good |
|             | link description: poor |
| e.g., MDL and B-NR |
| Overfitted  | link prediction: poor |
|             | link description: good |
| e.g., Q, Q-MR, and Infomap |
| Underfitted | link prediction: poor |
|             | link description: poor |
| e.g., B-HKK and Q-MP |
| Uneven      | overfits on some classes of inputs |
|             | underfits on others |
| e.g., spectral methods |

5.3 Discussion of Results

One consequence of the No Free Lunch and “no ground truth” theorems for community detection [19] is that algorithm evaluations based on comparing against a partition defined by node metadata do not provide generalizable or interpretable results. As a result, relatively little is known about the degree to which different algorithms over- or under-fit on or across different kinds of inputs.

The pair of link-based tasks introduced here defines a tradeoff much like a bias-variance tradeoff, in which an algorithm can either excel at learning the observed network (link description) or at learning to predict missing edges (link prediction), but not both. Link description thus represents a kind of in-sample learning evaluation, and an algorithm with a high benchmark performance curve on this task tends to correctly learn where edges do and do not appear in a given network. In contrast, link prediction presents a kind of out-of-sample learning evaluation, and an algorithm with a high performance curve on this task must do worse at link description in order to identify which observed non-edges are most likely to be missing links. The relative performance on these two tasks provides a qualitative diagnostic for the degree to which an algorithm tends to over- or under-fit when applied to real-world networks.

Box 3 provides simple guidelines for assessing this tendency for any particular method.

Our evaluation and comparison of 11 state-of-the-art algorithms under these two tasks using the CommunityFitNet corpus of 406 structurally diverse networks provides new insights. We summarize our findings in Table 2 describing the number of communities an algorithm tends to find, the group of algorithms its output partitions tend to be most similar to, and its performance on link prediction and description. We also provide an overall qualitative assessment of the algorithm’s practical behavior, following the rubrics in Box 3. In particular, we find dramatic differences in accuracy on both tasks across algorithms. Generally, we find that probabilistic methods perform well at link prediction and adequately at link description, indicating that they tend to produce relatively well-fitted communities.

In contrast, we find that methods based on modularity maximization (Q and Q-MR) and Infomap tend to overfit on real-world networks, performing poorly at link prediction.
A summary of results for 16 algorithms (Table 1) on the number of communities $k$ (Fig. 2b), the algorithm group the output is most similar to (Fig. 4), benchmark performance on link prediction (Fig. 5a) and link description (Fig. 5b), and an overall assessment of its tendency to over- or under-fit.

| Algorithm   | Number of Communities, $k$ | Partition Type       | Link Prediction Benchmark | Link Description Benchmark | Overall Fit       |
|-------------|-----------------------------|----------------------|---------------------------|---------------------------|------------------|
| Q           | larger                      | non-probabalistic    | poor                      | good                      | over fits        |
| Q-MR        | larger                      | non-probabalistic    | poor                      | good                      | over fits        |
| Q-MP        | smaller                     | spectral/ non-probabalistic | poor                      | poor                      | under fits       |
| Q-MP-G     | smaller                     | spectral/ non-probabalistic | —                         | —                         | inconclusive     |
| B-NR (SBM) | smaller                     | probabilistic        | very good                 | moderate                  | well-fitted      |
| B-NR (DC-SBM)| smaller                   | probabilistic        | moderate                  | very good                 | over fits, modestly |
| B-HKK (SBM) | smaller                     | probabilistic        | good                      | moderate                  | under fits, modestly |
| cICL-HKK (SBM) | smaller                   | probabilistic        | good                      | moderate                  | under fits, modestly |
| Infomap     | larger                      | non-probabalistic    | moderate                  | moderate                  | over fits        |
| MDL (SBM)   | smaller                     | probabilistic        | good                      | poor                      | under fits       |
| MDL (DC-SBM)| smaller                     | probabilistic        | very good                 | moderate                  | well-fitted      |
| S-NB        | smaller                     | spectral             | moderate                  | moderate                  | uneven fits      |
| S-cBHm      | smaller                     | spectral             | —                         | —                         | inconclusive     |
| S-cBHm      | smaller                     | spectral             | —                         | —                         | inconclusive     |
| AMOS        | larger                      | non-probabalistic    | —                         | —                         | inconclusive     |
| LRT-WB (DC-SBM)| larger                   | non-probabalistic    | —                         | —                         | inconclusive     |

Fig. 7. A heatmap showing the fraction of networks in the CommunityFitNet corpus on which a particular algorithm produced the best performance on the link prediction task, for different levels of subsampling $\alpha$. The two best overall methods (MDL DC-SBM and B-NR SBM) in Fig. 5a are not always the best, and every algorithm is the best for some combination of network and $\alpha$. Here, any algorithm with an AUC performance within 0.05 of the maximum observed AUC, for that network and $\alpha$ choice, is also considered to be “best”.

6 Conclusion

Community detection algorithms aim to solve the common problem of finding a lower-dimensional description of a complex network by identifying statistical regularities or patterns in connections among nodes. However, no algorithm can solve this problem optimally across all inputs [19], which produces a natural need to characterize the actual performance of different algorithms across different kinds of realistic inputs. Here, we have provided this characterization, for 16 state-of-the-art community detection algorithms applied to a large and structurally diverse corpus of real-world networks. The results shed considerable light on the
Fig. 8. Separate benchmark performance curves using model-specific score functions for the link prediction task for networks drawn from (a) biological (32%), (b) social (30%), (c) economic (27%), (d) technological (9%), (e) transportation (1%), and (f) information (1%) domains of origin in the CommunityFitNet corpus. As in Fig. 5a, each curve shows the mean AUC for a different community detection method, for a given fraction $\alpha$ of observed edges in a network.

broad diversity of behavior that these algorithms exhibit when applied to a consistent and realistic benchmark.

For instance, nearly all algorithms appear to find a number of communities that scales like $O(\sqrt{M})$. At the same time, however, algorithms can differ by more than an order of magnitude in precisely how many communities they find within the same network (Fig. 2). And, non-probabilistic approaches typically find more communities than probabilistic approaches. Comparing the precise composition of the identified communities across algorithms indicates that algorithms with similar underlying assumptions tend to produce similar kinds of communities—so much so that we can cluster algorithms based on their outputs (Fig. 4), with spectral techniques finding communities that are more similar to those found by other spectral techniques than to communities found by any other methods, and similarly for probabilistic methods and for non-probabilistic methods. This behavior would seem to indicate that different sets of reasonable assumptions about how to specify and find communities tend to drive real differences in the composition of the output. That is, different assumptions reflect different underlying tradeoffs, precisely as predicted by the No Free Lunch theorem.

Different algorithms also present wide variation in their tendency to over- or under-fit on real networks (Fig. 5), and the link prediction/description tasks we introduced provide a principled means by which to characterize this algorithmic tendency. Here also we find broad diversity across algorithms, with some algorithms, like MDL (DC-SBM) and B-NR (SBM) performing the best on average on link prediction and well enough on link description. However, we also find that these algorithms are not always the best at these tasks, and other algorithms can be better on specific networks (Fig. 7). This latter point is cautionary, as it suggests that comparative studies of community detection algorithms, which often rely on a relatively small number of networks by which to assess algorithm performance, are unlikely to provide generalizable results. The results of many previously published studies may need to be reevaluated in this light, and future studies may find the link prediction and link description tradeoff to be a useful measure of algorithm performance.

Beyond these insights into the algorithms themselves, the CommunityFitNet corpus of networks has several potential uses, e.g., it can be used as a standardized reference set for comparing community detection methods. To facilitate this use case, both the corpus dataset and the derived partitions for each member network by each of the algorithms evaluated here is available online for reuse. To compare a new algorithm with those in our evaluation set, a researcher can simply run the new algorithm on the corpus, and then identify which reference algorithm has the most similar behavior, e.g., in the average number of communities found (Fig. 2) or the composition of the communities obtained (Fig. 4). Similarly, a researcher could quickly identify specific networks for which their algorithm provides superior performance, as well as compare that performance on average across a structurally diverse set of
real-world networks. We expect that the availability of the CommunityFitNet corpus and the corresponding results of running a large number of state-of-the-art algorithms on it will facilitate many new and interesting advances in developing and understanding community detection algorithms.

Our results also open up several new directions of research in community detection. For instance, it would be valuable to investigate the possibility that a method, when applied to a single network, might over-partition some parts but under-partition other parts—an idea that could be studied using appropriate cross-validation on different parts of networks. Similarly, a theoretical understanding of what algorithmic features tend to lead to over- or under- or uneven-fitting outcomes for community detection would shed new light on how to control the underlying tradeoffs that lead to more general or more specific behavior. These tradeoffs must exist [19], and we find broad empirical evidence for them across our results here, but there is as yet no theoretical framework for understanding what they are or how to design around them for specific network analysis or modeling tasks.

### Appendix A

#### Model Selection Approaches

The general problem of choosing the number of communities $k$ is a kind of model selection problem, specifically, a kind of complexity control, as selecting more communities generally means more flexibility for the fitted model. Although it may be appealing to attempt to divide approaches based on whether $k$ is chosen explicitly as a parameter (as in many probabilistic approaches, like the SBM and its variants) or implicitly as part of the community detection itself (as in modularity maximization), such a dichotomy is not particularly clean. In this section, we survey the various different approaches to model selection in community detection.

Community detection methods can be divided in two broad categories: probabilistic and non-probabilistic methods. These two general groups cover roughly six classes of methods:

- Bayesian marginalization and regularized likelihood approaches [5], [7], [8].
- Information theoretic approaches [12].
- Modularity based methods [13].
- Spectral and other embedding techniques [9], [10], [11], [14], [15].
- Cross-validation methods [14], [15], and statistical hypothesis tests [13].

We note, however, that the boundaries among these classes are not rigid and one method can belong to more than one group. For example, MDL is both an information theoretic approach as well as a Bayesian approach, and modularity can be viewed as a special case of the DC-SBM.

Many probabilistic approaches choose a parametric model like the popular SBM or one of its variants, and then design specific rules for model selection (choosing $k$). One principled way to avoid overfitting is to use the minimum description length (MDL) [57] method, which tries to compress the data via capturing its regularities. Ref. [20] employs MDL on networks and aims to avoid overfitting via trading off the goodness of fit on the observed network with the description length of the model. This approach can also be generalized to hierarchical clustering and overlapping communities [56], [58].

The probabilistic group includes the Regularized-Likelihood approaches [8], [59], [28] and Bayesian model selection methods [6], [29], [7], [31]. Regularized-Likelihood approaches are similar to Bayesian Information Criterion (BIC) in model selection [60]. Ref. [61] proposes to select the number of clusters in mixture models, using some criterion called the integrated complete likelihood (ICL) instead of BIC. Basically, BIC does not take into account the entropy of the fuzzy classification and ICL is intended to find more reliable clusters by adding this entropy into the penalty terms. However, computing ICL in the setting of a network mixture model like the SBM is not tractable. To address this issue for the SBM, Ref. [8] proposed using ICL and approximating it by resorting to the frequentist variational EM. Because of asymptotic approximations in ICL, these results are not reliable for smaller networks. In another study [59], the authors employ variational Bayes EM and propose to use the ILvb criterion for complexity control. In both the ICL and ILvb [8], [59] approaches, some approximations are used. Ref. [62] bypasses these approximations by considering the conjugate priors and tries to improve the results by finding an analytical solution. Also in Ref. [28], the authors find the exact ICL by using an analytical expression and propose a greedy algorithm to find the number of clusters and partition the network simultaneously.

We categorize regularized-likelihood and Bayesian approaches together, because the prior beliefs in Bayesian approaches play a similar role to penalty terms in penalized likelihood functions. Bayesian marginalization and related approaches aim to control for overfitting by averaging over different parameterizations of the model. The various approaches in the Bayesian group use different approximations in order to make this averaging computationally feasible in a network setting. A common practice for networks, e.g., starting with the SBM, is to either use a Laplace approximation or use conjugate priors [29], [7], [31], both of which yield a penalty term that can be compared with penalty terms in regularized methods. Different choices in the particular priors [29], [7] or in the order of Laplace approximations [29], [31] yield different resulting model selection specifications. Similarly, Ref. [29] chooses a maximum entropy prior (B-NR), while Ref. [7] chooses a uniform prior.

An approximation technique known as factorized information criterion (FIC) is explored in the context of networks in Ref. [31], along with its corresponding inference method known as factorized asymptotic Bayesian (FAB). Ref. [31] adapts this criterion to the SBM and name it $F^2$IC, which is more precise than FIC and is specifically designed for SBM. A tractable algorithm named $F^2$AB (B-HKK) is proposed to carry out Bayesian inference. A key property of the FIC is that it can account for dependencies among latent variables and parameters, and is asymptotically consistent.
with the marginal log-likelihood. Ref. [31] also proposes a modification to the ICL criterion [8] that corresponds to the simplified version of FIC [53], and which is referred to as corrected ICL (cICL-HKK) here.

In contrast to the description length approaches taken with probabilistic models like the SBM, Ref. [12] proposes a different information theoretic approach known as Infomap, which uses compression techniques on the paths of a random walker to identify community structure regularities in a network. This approach can be generalized to hierarchical community structure [35] and to overlapping modular organization [64].

In modularity based methods [5, 24], an objective function based on a particular goodness of fit measure is proposed and the corresponding optimization over partitions can be solved in any number of ways. Undoubtedly, the most widespread measure in this category is modularity Q proposed by Newman and Girvan [5]. Modularity maximization favors putting the nodes with large number of connections inside the clusters compared to the expected connections under a random graph with the same degree sequence.

Recently, Ref. [30] showed that multiresolution modularity (Q-MR) maximization is mathematically equivalent to a special case of the DC-SBM, under a k-planted partition parameterization. The Q-MR algorithm works implicitly like a likelihood maximization algorithm, except that it chooses its resolution parameter, which sets the number of communities k, by iterating between the Q and DC-SBM formulations of the model. In another modularity based approach [24], the authors propose a message passing algorithm (Q-MP) by introducing a Gibbs distribution utilizing the modularity as the Hamiltonian of a spin system, and a means for model selection via minimization of the Bethe free energy. This approach enables marginalization over the ruggedness of the modularity landscape, providing a kind of complexity control not available through traditional modularity maximization. The main issue is that to infer informative communities, some parameters of the model (inverse temperature $\beta$) need to be chosen so that the model does not enter the spin-glass phase. Ref. [17] builds on this approach by proposing a generalized version of modularity message passing (Q-GMP) for model selection that infers the parameters of Boltzmann distribution (inverse temperature $\beta$) instead of just setting it to some pre-calculated value.

Spectral methods using eigen decomposition techniques can find the informative eigenvectors of an adjacency matrix or a graph Laplacian, and the embedded coordinates can be used for community detection. However, traditional spectral approaches are not appropriate in clustering sparse networks, or networks with heavy-tailed degree distributions. Recently, Ref. [9] proposed a spectral approach for community detection in sparse networks based on the non-backtracking matrix (S-NB), that succeeds all the way down to the detectability limit in the stochastic block model [49]. In this setting, the number of communities k is chosen by the number of real eigenvalues outside the spectral band. More recently, Ref. [10] proposes to choose k as the number of negative eigenvalues of the Bethe Hessian matrix. Ref. [11] proves the consistency of these approaches in dense and sparse regimes and also describes some corrections on the spectral methods of Refs. [9] and [10] (S-cBHm and S-cBHa).

There is another venue of embedding techniques used in clustering, related to feature learning in networks. Following the recent achievements in natural language processing via the skip-gram model, Ref. [55] develops an algorithm to encode a representation of graph vertices by modeling and then embedding a stream of rigid random walks. Ref. [54] generalizes this idea and proposes an algorithm to learn continuous feature representations for nodes, which can be used in community detection and for learning which nodes have similar structure. Two attractive properties of such node-embedding approaches are their scalability and the ease with which they can be used to make predictions about edges. These methods are not included in our study as they have not yet been well explored in the context of community detection.

Traditional approaches to evaluating and controlling for model overfit, such as optimizing the bias variance tradeoff, fail in network settings because pairwise interactions violate standard independence assumptions. Because of this non-independence issue, cross-validation techniques are not theoretically well developed in the context of networks, and even simple edge-wise cross-validation can be computationally expensive. Recently, Ref. [15] showed that the leave-one-out cross-validation prediction error can be efficiently computed using belief propagation (BP) in sparse networks and thereby efficiently used for model selection. Similarly, Ref. [14] estimates the number of communities using a block-wise node-pair cross-validation method, which can be adapted to any other algorithm and model. The number of communities is chosen by validating on the testing set (minimizing the generalization error) and the technique can simultaneously chooses between SBM and DC-SBM by selecting the minimum validation error. However, it should be noted that recently Ref. [25] showed that model selection techniques based on cross-validation are not always consistent with the most parsimonious model and in some cases can lead to overfitting.

Statistical methods test the number of clusters using some test statistics through a recursive hypothesis testing approach. In general, these approaches have a high computational complexity because of this outer loop. Ref. [32] proposes an algorithm for automated model order selection (AMOS) in networks for random interconnection model (RIM) (a generalization of the SBM). The method uses a recursive spectral clustering approach, which increases the number of clusters and tests the quality of the identified clusters using some test statistics achieved by phase transition analysis. Ref. [32] proves this approach to be reliable under certain constraints. Ref. [13] proposes a likelihood ratio test (LRT-WB) statistic for the SBM or DC-SBM to choose the number of clusters k, and shows that when the average degree grows poly-logarithmically in N, the correct order of a penalty term in a regularized likelihood scheme can be derived, implying that its results are asymptotically consistent.

Ref. [65] proposes a sequential hypothesis testing approach to choose k. At each step, it tests whether to bipartition a graph or not. To this end, the authors derive and utilize the asymptotic null distribution for Erdős-Rényi random graphs. This possibility originated from the fact that
the distribution of the leading eigenvalue of the normalized adjacency matrix under the SBM converges to the Tracy-Widom distribution. Ref. [66] uses recent results in random matrix theory to generalize the approach of Ref. [65] to find the null distribution for SBMs in a more general setting. Utilizing this null distribution and using the test statistic as the largest singular value of the residual matrix, computed by removing the estimated block model from the adjacency matrix, Ref. [65] proposes an algorithm to choose $k$ by testing $k = k_0$ versus $k > k_0$ sequentially for each $k_0 \geq 1$, which is proved to be consistent under a set of loose constraints on the number of clusters ($k = O(N^{1/6} - \tau)$ for some $\tau > 0$) and the size of clusters ($\Omega(N^{5/6})$).

It is noteworthy that Ref. [42] recently proved that one constraint on the sizes of inferred communities under some methods is an artifact of identifying the large and small clusters simultaneously. It goes on to show that this issue can be resolved using a technique called “peeling,” which first finds the larger communities and then, after removing them, finds the smaller-sized communities using appropriate thresholds. This iterative approach is similar to the superposition coding technique in coding theory and recalls the hierarchical clustering strategy introduced in Ref. [35] for capturing clusters with small sizes. Basically, by iteratively limiting the search space, finding an optimum solution becomes computationally more tractable. Relatedly, Ref. [43] shows that in planted $k$-partition model, the space of parameters of the model divides into four regions of impossible, hard, easy and simple, which are related to the regimes that algorithms based on maximum likelihood estimators can succeed theoretically and/or computationally. These results indicate that no computationally efficient parametric algorithm can find clusters if the number of clusters increase unbounded over $\Omega(\sqrt{N})$. This fact is in strong agreement with our experimental results.

ACKNOWLEDGMENTS

The authors thank Tiago Peixoto, Leto Peel, and Daniel Larremore for helpful conversations, and acknowledge the BioFrontiers Computing Core at the University of Colorado Boulder for providing High Performance Computing resources (NIH 1S10OD012300) supported by BioFrontiers IT. The authors also thank Mark Newman, Rachel Wang, Peter Bickel, Can Le, Elizaveta Levina, Tatsuro Kawamoto, Kohei Hayashi, Pin-Yu Chen, and Etienne Côme for sharing their codes. The authors thank Ellen Tucker for providing the network dataset from ICON. Financial support for this research was provided in part by Grant No. IIS-1452718 (AG, AC) from the National Science Foundation.

REFERENCES

[1] M. A. Porter, J.-P. Onnela, and P. J. Mucha, “Communities in networks,” Notices of the American Mathematical Society, vol. 56, no. 9, pp. 1082–1097, 1164–1166, 2009.

[2] S. Fortunato, “Community detection in graphs,” Physics Reports, vol. 486, pp. 75–174, 2010.

[3] T. P. Peixoto, “Bayesian stochastic blockmodeling,” arXiv:1705.0225, 2017.

[4] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” Soc. Netw., vol. 5, no. 2, pp. 109–137, 1983.

[5] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” Phys. Rev. E, vol. 69, no. 2, p. 026113, 2004.

[6] J. M. Hofman and C. H. Wiggins, “Bayesian approach to network modularity,” Phys. Rev. Lett., vol. 100, no. 25, p. 258701, 2008.

[7] X. Yan, “Bayesian model selection,” in IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2016, pp. 323–328.

[8] J.-J. Daudin, F. Picard, and S. Robin, “A mixture model for random graphs,” Stat. Comput., vol. 18, no. 2, pp. 173–183, 2008.

[9] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, “Spectral redemption in clustering sparse networks,” Proc. Natl. Acad. Sci., vol. 110, no. 52, pp. 20935–20940, 2013.

[10] A. Saade, F. Krzakala, and L. Zdeborová, “Spectral clustering of graphs with the Bethe Hessian,” in Adv. Neural Info. Proc. Sys., 2014, pp. 406–414.

[11] C. M. Le and E. Levina, “Estimating the number of communities in networks by spectral methods,” arXiv:1507.00827, 2015.

[12] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks,” Proc. Natl. Acad. Sci. USA, vol. 105, no. 4, pp. 1118–1123, 2008.

[13] Y. R. Wang, P. J. Bickel et al., “Likelihood-based model selection for stochastic block models,” Ann. Stat., vol. 45, no. 2, pp. 500–528, 2017.

[14] K. Chen and J. Lei, “Network cross-validation for determining the number of communities in network data,” J. Am. Stat. Assoc., no. just-accepted, 2016.

[15] T. Kawamoto and Y. Kabashima, “Cross-validation estimate of the number of clusters in a network,” Sci. Rep., vol. 7, 2017.

[16] D. Hric, R. K. Darst, and S. Fortunato, “Community detection in networks: Structural communities versus ground truth,” Phys. Rev. E, vol. 90, p. 062805, 2014.

[17] T. Kawamoto and Y. Kabashima, “Comparative analysis of model selection and link prediction for stochastic block models,” arXiv:1605.07668, 2016.

[18] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” Physics Reports, vol. 659, pp. 1–44, 2016.

[19] L. Peel, D. B. Larremore, and A. Clauset, “The ground truth about metadata and community detection in networks,” Sci. Adv., vol. 3, no. 5, p. e1602548, 2017.

[20] T. P. Peixoto, “ Parsimonious module inference in large networks,” Phys. Rev. Lett., vol. 110, no. 14, p. 148701, 2013.

[21] F. Zhang, F. Krzakala, J. Reichardt, and L. Zdeborová, “Comparative study for inference of hidden classes in stochastic block models,” J. Stat. Mech. Theor. Exp., vol. 2012, no. 12, p. P12021, 2012.

[22] P. Hoff, “Modeling homophily and stochastic equivalence in symmetric relational data,” in Adv. Neural Info. Proc. Sys., 2008, pp. 657–666.

[23] B. Dabb and B. Junker, “Comparison of cross-validation methods for stochastic block models,” arXiv:1605.03000, 2016.

[24] P. Zhang and C. Moore, “Scalable detection of statistically significant communities and hierarchies, using message passing for modularity,” Proc. Natl. Acad. Sci. USA, vol. 111, no. 51, pp. 18144–18149, 2014.

[25] T. Vallés-Català, T. P. Peixoto, R. Guimerà, and M. Sales-Pardo, “On the consistency between model selection and link prediction in networks,” arXiv:1705.07967, 2017.

[26] B. Karrer and M. E. J. Newman, “Stochastic blockmodels and community structure in networks,” Phys. Rev. E, vol. 83, no. 1, p. 016107, 2011.

[27] A. Condon and R. M. Karp, “Algorithms for graph partitioning on just-accepted,” 2001.

[28] E. Côme and P. Latouche, “Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood,” Stat. Model., vol. 15, no. 6, pp. 564–589, 2015.

[29] M. E. J. Newman and G. Reinert, “Estimating the number of communities in a network,” Phys. Rev. Lett., vol. 117, no. 7, p. 078301, 2016.

[30] M. E. J. Newman, “Community detection in networks: Modularity optimization and maximum likelihood are equivalent,” arXiv:1606.02319, 2016.

[31] K. Hayashi, T. Konishi, and T. Kawamoto, “A tractable fully Bayesian method for the stochastic block model,” arXiv:1602.02256, 2016.
[32] P.-Y. Chen and A. O. Hero, “Phase transitions and a model order selection criterion for spectral graph clustering,” arXiv:1604.03159, 2016.

[33] M. Sales-Pardo, R. Guimerà, A. A. Moreira, and L. A. N. Amaral, “Extracting the hierarchical organization of complex systems,” Proc. Natl. Acad. Sci. USA, vol. 104, no. 15, pp. 5224–5229, 2007.

[34] A. Clauset, C. Moore, and M. E. J. Newman, “Hierarchical structure and the prediction of missing links in networks,” Nature, vol. 453, pp. 98–101, 2008.

[35] M. Rosvall and C. T. Bergstrom, “Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems,” PloS one, vol. 6, no. 4, p. e18209, 2011.

[36] T. Peixoto, “Hierarchical block structures and high-resolution model selection in large networks,” Phys. Rev. X, vol. 4, no. 1, p. 011047, 2014.

[37] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” J. Mach. Learn. Res., vol. 9, no. Sep, pp. 1981–2014, 2008.

[38] A. Clauset, E. Tucker, and M. Sainz, “The Colorado Index of Complex Networks,” https://icon.colorado.edu/, 2016.

[39] Y. Chen, S. Sanghavi, and H. Xu, “Improved graph clustering,” J. Mach. Learn. Res., vol. 16, pp. 455–490, 2015.

[40] A. Clauset, A. A. Moreira, and L. A. N. Amaral, “Missing and spurious interactions and the reconstruction of complex networks,” Proc. Natl. Acad. Sci. USA, vol. 106, no. 52, pp. 22 073–22 078, 2009.

[41] R. Guimerà and M. Sales-Pardo, “Asymptotic inference and complexity control for stochastic block models,” Statistical Modelling, vol. 12, no. 1, pp. 93–115, 2012.

[42] A. Clauset, E. Tucker, and M. Sainz, “The Colorado Index of Complex Networks,” https://icon.colorado.edu/, 2016.