Community Detection through Likelihood Optimization: In Search of a Sound Model

Liudmila Prokhorenkova
Moscow Institute of Physics and Technology
Yandex
Moscow, Russia
ostroumova-la@yandex.ru

Alexey Tikhonov
Yandex
Berlin, Germany
altsoph@gmail.com

ABSTRACT

Community detection is one of the most important problems in network analysis. Among many algorithms proposed for this task, methods based on statistical inference are of particular interest: they are mathematically sound and were shown to provide partitions of good quality. Statistical inference methods are based on fitting some random graph model (a.k.a. *null model*) to the observed network by maximizing the likelihood. The choice of this model is extremely important and is the main focus of the current study. We provide an extensive theoretical and empirical analysis to compare several models: the widely used planted partition model, recently proposed degree-corrected modification of this model, and a new null model having some desirable statistical properties. We also develop and compare two likelihood optimization algorithms suitable for the models under consideration. An extensive empirical analysis on a variety of datasets shows, in particular, that the new model is the best one for describing most of the considered real-world complex networks according to the likelihood of observed graph structures.

CCS CONCEPTS

• **Mathematics of computing** → **Random graphs**; **Graph algorithms**; **Maximum likelihood estimation**; • **Theory of computation** → **Random network models**.

KEYWORDS

Community detection; likelihood optimization; statistical inference; planted partition model; LFR benchmark

1 INTRODUCTION

Among various properties shared by many real-world complex networks, community structure is extremely important. It is characterized by the presence of highly interconnected groups of vertices (*communities* or *clusters*) relatively well separated from the rest of

the network. In social networks communities are formed by users with similar interests, in citation networks they represent papers on related topics, etc. The presence of communities highly affects, e.g., the promotion of products via viral marketing, the spreading of infectious diseases, computer viruses and information, and so on. Being able to identify communities in a network could help us to exploit this network more effectively: find similar scientific papers, discover users with similar interests for targeted advertisement, compress or visualize the network, etc.

Among other algorithms proposed for community detection, the notable ones are methods based on statistical inference. In such methods, some underlying random graph model is assumed, the evidence is represented by the observed graph, and hidden variables to be inferred are the parameters of the model together with community assignments. Such methods are appealing since they are theoretically sound and consistent: e.g., it has been proved that when the maximum-likelihood method is applied to networks generated from the same stochastic block model, it returns correct cluster assignments in the limit of large degrees [7]. Also, likelihood can be used to formalize the notion of a community [15].

The choice of the proper null model is essential for statistical inference algorithms as it highly affects their performance. There are two types of models: with fixed or non-fixed number of parameters. Direct likelihood optimization for models with non-fixed number of parameters leads to trivial partitions, as discussed in Section 4.1, and additional heuristics are needed to define the number of clusters. Hence, in this paper we focus on models with fixed number of parameters. We analyze strengths and weaknesses of two most popular models — planted partition model and its degree-corrected variant (they are discussed in Section 3). We also propose a new model, which has only one parameter, satisfies a desired statistical property of preserving the expected degree sequence, and shows the best fit for a variety of real-worlds networks according to the likelihood of observed structures. To the best of our knowledge, we are the first to perform a thorough theoretical and empirical evaluation of the null models used for community detection.

To sum up, the contributions of this paper are the following:

- We theoretically analyze two well-known null models used for community detection. Based on this analysis, we propose a new one-parameter model which has a desirable statistical property of preserving the expected degree sequence.
- We empirically demonstrate that the new model gives the best fit to a variety of real-world complex networks.
- We show that the likelihood based on the proposed model can be used as a new, more reliable, alternative to the well-known modularity measure.

- We develop several likelihood optimization algorithms suitable for all discussed models. We perform an extensive empirical comparison of all the algorithms on various synthetic and real-world datasets.
- The implementation of the proposed algorithms is available as an open-source library.[1]

The rest of the paper is organized as follows. In the next section we overview related papers and introduce the required background. Then, in Section 3, we formally define and analyze all null models used in the current research. In Section 4, we discuss the method of statistical inference and present the likelihoods for all models. The proposed likelihood optimization algorithms are discussed in Section 5. Section 6 provides an extensive empirical analysis of all models and algorithms and Section 7 concludes the paper.

2 BACKGROUND AND RELATED WORK

2.1 Community Detection Methods

In this section, we briefly overview algorithms used for community detection. However, most of the related work will be discussed in the corresponding background sections: Section 2.2 discusses modularity measure; Section 2.3 defines the LFR bechmark; Sections 3.1 and 3.2 describe standard null models; Section 4.2 discusses papers on statistical inference; Section 5.1 deals with some optimization techniques. For a deeper overview of community detection area, we refer the reader to several survey papers, e.g., [10, 16, 19, 21, 33].

The problem of community detection has recently attracted a lot of attention and many algorithms were proposed to solve this problem. The proposed methods include: spectral algorithms [55]; methods based on statistical inference (see Section 4); methods based on optimization (see Section 5); methods based on dynamical processes on networks, like diffusion, spin dynamics, synchronization, and so on. Also, existing approaches differ by a particular task at hand: detecting overlapping communities [57], clustering in dynamic [51], directed [48], or weighted [3] networks, community detection in annotated [39] or layered [44] networks, and so on.

In this paper, we focus on the basic problem of detecting non-overlapping communities in undirected and unweighted graphs. However, ideas discussed in this paper can be applied to more general cases, since there exist the corresponding generalizations of the null models. We leave this analysis for the future research, since there is a lot to be understood even for the basic case.

2.2 Modularity

Throughout this paper we assume that we are given a simple undirected graph $G = (V(G), E(G))$ and the task is to find a partition $C = \{C_1, \ldots, C_k\}$ of its vertex set $V(G)$ such that $\bigcup C_i = V(G)$ and $C_i \bigcap C_j = \emptyset$ for $i \neq j$. The notation used in the paper is summarized in Table 1.

Let us introduce the notion of *modularity*, which is the most well-known measure of a goodness of a partition C for a graph G. It was first introduced in [40] and is widely used in community detection literature: modularity is directly optimized by some algorithms, used as a stopping criteria in iterative methods, or used as a metric to compare different algorithms when no ground truth

[1]https://github.com/altsoph/community_loglike

Table 1: Notation

Variable	Description
G	observed graph, simple and undirected
$V(G)$	set of vertices
$E(G)$	set of edges
$n = |V(G)|$	number of vertices in G
$m = |E(G)|$	number of edges in G
$C = \{C_1, \ldots, C_k\}$	partition of $V(G)$
$k = |C|$	number of communities
$Q_0(C, G, \gamma)$	simple modularity defined in (1)
$Q_1(C, G, \gamma)$	standard modularity defined in (2)
$c(i)$	cluster assignment of vertex i
$m_{in} = m_{in}(C, G)$	number of intra-community edges
$m_{out} = m_{out}(C, G)$	number of inter-community edges
$P = \binom{n}{2}$	number of pairs of vertices
$P_{in} = \sum_{C \in C} \binom{|C|}{2}$	number of intra-community pairs of vertices
$P_{out} = P - P_{in}$	number of inter-community pairs of vertices
$d(i)$	degree of vertex i in G
$d_{in}(i)$	number edges going from i to vertices in $C_{c(i)}$
$d_{out}(i) = d - d_{in}(i)$	number of edges going from i to outside $C_{c(i)}$
$D(C) = \sum_{i \in C} d(i)$	degree of cluster C
$D_{in}(C) = \sum_{i \in C} d_{in}(i)$	twice the number of edges induced by C
$m(C_q, C_r)$	number of edges between C_q and C_r or twice the number of intra-cluster edges if $q = r$

partition is available. The basic idea is to consider the fraction of intra-community edges among all edges of G and penalize it in order to avoid trivial partitions like $C = \{V(G)\}$. In its general form, modularity is

$$Q(C, G, \gamma) = \frac{1}{m} \left(m_{in} - \gamma \mathbb{E}(m'_{in}) \right),$$

where $m = m(G) = |E(G)|$ is the number of edges in G; $m_{in} = m_{in}(C, G)$ is the number of intra-community edges; $m'_{in} = m'_{in}(C)$ is a (random) number of intra-community edges in a graph constructed according to some underlying random graph model; γ is the *resolution parameter* [47], which is usually added to get the possibility of varying the number of communities obtained after maximizing $Q(C, G, \gamma)$.

The simplest underlying model is the Erdős–Rényi random graph, i.e., we assume that G is sampled uniformly at random from the set of all graphs with n vertices and m edges. For this model, $\mathbb{E}(m'_{in}) = P_{in} \cdot \frac{m}{P}$, where $P = \binom{n}{2}$ is the number of pairs of vertices and $P_{in} = \sum_{C \in C} \binom{|C|}{2}$ is the number of intra-community pairs. So, the formula for *simple modularity* is

$$Q_0(C, G, \gamma) = \frac{1}{m} \left(m_{in} - \gamma \frac{P_{in} m}{P} \right). \tag{1}$$

However, the Erdős–Rényi random graph model is known to be not a good descriptor of real-world networks since its Poisson degree distribution significantly differs from heavy-tailed degree distributions observed in real-world networks. A standard solution is to consider a random graph with a given expected degree sequence [12] as a null model, and to take the degree sequence

from the observed graph.[2] For this model, the probability that vertices i and j are connected equals $\frac{d(i)d(j)}{2m}$, where $d(i)$ is the degree of a vertex i.[3] In this case we have $\mathbb{E}(m'_{in}) = \frac{1}{4m} \sum_{C \in \mathcal{C}} D(C)^2$, where $D(C) = \sum_{i \in C} d(i)$ is the degree of cluster C. So, the standard expression for *modularity* is

$$Q_1(\mathcal{C}, G, \gamma) = \frac{m_{in}}{m} - \frac{\gamma}{4m^2} \sum_{C \in \mathcal{C}} D(C)^2 . \tag{2}$$

Originally, modularity was introduced with $\gamma = 1$, but is was proven to have a resolution limit [20], i.e., it fails to detect communities smaller than a certain size; varying the resolution parameter helps to overcome this problem, but tuning γ is challenging.

Despite some known shortcomings [19], modularity remains the most popular way to measure the quality of a partition and many algorithms are based on direct modularity optimization (see Section 5). Also, under certain assumptions, modularity maximization is equivalent to likelihood optimization [36].

2.3 LFR Model

Let us now describe the LFR (Lancichinetti-Fortunato-Radicchi) model [29], which is the most widely used synthetic benchmark for comparison of community detection algorithms.

LFR generates a graph G on n vertices. The main parameters of the model are: the exponent of the power-law degree distribution γ, the exponent of the power-law community size distribution β, the average degree $\bar{d}$, and mixing parameter $\hat{\mu}$. First, we generate the degrees of vertices by sampling them independently from the power-law distribution with exponent γ and mean $\bar{d}$. Each vertex shares a fraction $1 - \hat{\mu}$ of its edges with the vertices of its community and a fraction $\hat{\mu}$ with the other vertices of the network. The sizes of the communities are sampled from the power-law distribution with exponent β, such that the sum of all sizes equals n. Then, vertices are assigned to communities such that the internal degree of any vertex is less than the size of its community. Finally, the configuration model [35] with rewiring steps is used to construct a graph with the given degree sequence and with the required fraction of internal edges. See [29] for the detailed description of this procedure.

3 NULL MODELS

In this section, we analyze several random graph models having community structure and suitable for using as null models for likelihood optimization.

3.1 Stochastic Block Model

The most well-known random graph model with community structure is the *stochastic block model* (SBM) [26]. In this model, the vertices are divided in k clusters and for each pair of vertices i, j we draw an edge between them with probability $p_{c(i),c(j)}$ independently of all other edges. Here $c(i)$ is a community assignment for a

vertex i. The values $p_{q,r}$, $1 \le q, r \le k$ are parameters of the model, $0 \le p_{q,r} \le 1$ and $p_{q,r} = p_{r,q}$. In many community detection algorithms a simplified version of SBM called *planted partition model* (PPM) is used [14]. PPM is a special case of SBM, where $p_{q,q} = p_{in}$ for $1 \le q \le k$ and $p_{q,r} = p_{out}$ for $1 \le q, r \le k, q \ne r$.

3.2 Degree-Corrected Stochastic Block Model

Similarly to the Erdős–Rényi random graph, SBM is unable to model heavy-tailed degree distributions, so community detection algorithms based on this model may have a poor quality. To overcome this issue, Karrer and Newman proposed the *degree-corrected stochastic block model* (DCSBM) [28]. In this model the vertices are again assigned to k clusters and edges are placed independently at random. The number of edges between vertices i and j follows a Poisson distribution with the mean $\frac{d(i)d(j)}{2m} p_{c(i),c(j)}$ or a half that number for self-loops. Let us show that DCSBM is able to generate graphs with desired expected degrees.

PROPOSITION 1. *There exist such $p_{q,r}$ that in DCPPM we have $\mathbb{E}(d'(i)) = d(i)$ for all i.*

PROOF. Let us set $p_{q,r} = \frac{2\,m\,m(C_q, C_r)}{D(C_q)\,D(C_r)}$, where $m(C_q, C_r)$ is the number of edges between clusters C_q and C_r or twice the number of intra-cluster edges if $q = r$. Note that these parameters maximize the likelihood for DCSBM [28]. For such parameters we get the desired condition:

$$\mathbb{E}(d'(i)) = \sum_{j=1}^{n} \frac{d(i)d(j)}{2m} p_{c(i),c(j)}$$

$$= \frac{d(i)}{D(C_{c(i)})} \sum_{C \in \mathcal{C}} \frac{m(C, C_{c(i)})}{D(C)} \sum_{j \in C} d(j) = d(i) .$$

So, we proved that in its general form DCSBM is able to preserve the desired degree sequence. □

The *degree-corrected planted partition model* (DCPPM) [36] is a simplified version of DCSBM, where $p_{q,q} = p_{in}$ for $1 \le q \le k$ and $p_{q,r} = p_{out}$ for $1 \le q, r \le k, q \ne r$. Let us prove the following "negative" proposition.

PROPOSITION 2. *If $p_{in} \ne p_{out}$ and for some q, r we have $D(C_q) \ne D(C_r)$, then $\mathbb{E}(d'(i)) \ne d(i)$ for some i.*

PROOF. Let us compute the expected degree of i:

$$\mathbb{E}(d'(i)) = \frac{d(i)}{2m} p_{in} \sum_{j \in C_{c(i)}} d(j) + \frac{d(i)}{2m} p_{out} \sum_{C \ne C_{c(i)}} \sum_{j \in C} d(j)$$

$$= \frac{d(i)}{2m} \left(D(C_{c(i)})(p_{in} - p_{out}) + 2m\,p_{out} \right) .$$

To get $\mathbb{E}(d'(i)) = d(i)$ for all i, we need to have, for *any* cluster C, $D(C)(p_{in} - p_{out}) + 2m\,p_{out} = 2m$. As we have at least two clusters with different values $D(C)$, we have to take $p_{in} = p_{out} = 1$, which leads to the standard configuration model without any community structure and contradicts the statement of the proposition. □

As a result, we obtain that the standard method of limiting the number of parameters in DCSBM leads to inability of the obtained model to preserve the expected degree sequence.

---

[2]At large scale, this model is essentially equivalent to the configuration model [35]: a random graph sampled uniformly from the set of graphs with a given degree sequence.
[3]For technical purposes, it is more convenient (and conventional) to allow the underlying model to have loops and multiple edges and to assume that the number of edges between i and j follows a Poisson distribution with the mean $\frac{d(i)d(j)}{2m}$ if $i \ne i$ and the expected number of loops for a vertex i is $\frac{d(i)^2}{4m}$. Note that usually $d(i)d(j) \ll 2m$, so multiple edges rarely appear.

3.3 Independent LFR Model

Motivated by Proposition 2, we developed a one-parameter model which preserves the expected degree sequence. It is a special case of DCSBM and we call it *independent LFR model* (ILFR) due to its analogy to the LFR benchmark.

The core problem of DCPPM is the assumption that the probability of an internal edge is independent of the size of its community, so the fraction of internal edges for a vertex *depends on the community size*. On the contrary, we propose using the mixing parameter μ to control this fraction. Namely, we consider DCSBM with the following expected number of edges between two vertices i and j: $\frac{\mu d(i)d(j)}{2m}$ if $c(i) \neq c(j)$ or $\frac{(1-\mu)d(i)d(j)}{D(C_{c(i)})} + \frac{\mu d(i)d(j)}{2m}$ if $c(i) = c(j)$ (or half this number for the self loops). Note that making the probability of an internal edge dependent on the size of the community is very natural: e.g., if a community of people is very small, one would expect that its members are much closer related to each other than members of large communities. The following proposition holds for the proposed model.

PROPOSITION 3. *In ILFR we have* $\mathbb{E}(d'(i)) = d(i)$ *for all* i.

PROOF. Indeed, let us compute the expected degree of i:

$$\mathbb{E}(d'(i)) = d(i)\left((1-\mu)\sum_{j \in C_{c(i)}} \frac{d(j)}{D(C_{c(i)})} + \mu \sum_{j \in V(G)} \frac{d(j)}{2m}\right) = d(i)\,.$$

$\square$

So, this model solves the problem with expected degrees. Another advantage of ILFR is that it has only one parameter μ instead of p_{in} and p_{out} in planted partition models.

4 STATISTICAL INFERENCE

4.1 Method

Having discussed possible options for the null models, we are ready to describe the statistical inference method. For community detection problem, this method consists of two ingredients: the evidence, expressed by a graph G, and a random graph null model with parameters θ (θ include community assignments and μ or p_{in} and p_{out} for the models under consideration). The goal is to find such parameters θ that maximize the posterior distribution

$$\mathrm{P}(\theta|G) = \frac{\mathrm{P}(G|\theta)\mathrm{P}(\theta)}{\mathrm{P}(G)}\,,$$

where $\mathrm{P}(\theta)$ is the prior distribution of parameters and $\mathrm{P}(G) = \int \mathrm{P}(G|\bar{\theta})\mathrm{P}(\bar{\theta})\,d\bar{\theta}$ is independent of θ, therefore $\arg\max_\theta \mathrm{P}(\theta|G) = \arg\max_\theta \mathrm{P}(G|\theta)\mathrm{P}(\theta)$. The choice of the prior distribution $\mathrm{P}(\theta)$ is usually not obvious. Therefore, in the community detection literature, the likelihood $P(G|\theta)$ that the model is consistent with the observed graph structure is often maximized.

Let us also stress that in Section 3 we discussed two types of models: SBM/DCSBM having non-fixed number of parameters and PPM/DCPPM/ILFR with a fixed number of parameters. If the number of parameters is fixed, then the likelihood optimization can be applied directly. Otherwise, one needs another tool to find the optimal number of clusters, since direct likelihood maximization would lead to a trivial partition with all vertices forming their own clusters and the probability of an edge between two vertices equals

1 if they are connected and 0 otherwise. To avoid this problem, we further focus only on the models with fixed number of parameters.

4.2 Related Work on Statistical Inference

Let us briefly describe previous literature on statistical inference in community detection. Hastings [24] considers PPM, as defined in Section 3.1. He shows a connection between the obtained likelihood and the Hamiltonian of a Potts model with short- and long-range interactions. For $p_{in} > p_{out}$, the model is a spin glass with ferromagnetic nearest-neighbor interactions and antiferromagnetic long-range interactions. Belief propagation is used to find the ground state of the spin model. Another method based on the directed SBM is presented by Newman and Leicht [41], they use the expectation-maximization technique to optimize the parameters. These ideas are close to the a posteriori blockmodel proposed in [42]. A variational approach to parameter estimation is adopted in [17]. Some improvements and modifications of these approaches are introduced in, e.g., [11, 25, 45, 59]. A generalization to the overlapping SBM is proposed by McDaid and Hurley [34]. As we discussed, DCSBM is defined in [28] and communities are detected based on the likelihood in this model. Copic et al. [15] define an axiomatization for the problem of community detection based on PPM. Likelihood is used as a quality function to define rankings between graph partitions and it is shown that such rankings satisfy a number of desired properties. The authors also propose an algorithm to find (approximately) the maximum likelihood partition.

Finally, if the number of parameters in the model is not fixed, some additional tools are required to figure out the optimal number of clusters. Here the average entropy of the classification [45] or the Integrated Classification Likelihood [17, 59] can be used.

4.3 Likelihood for PPM and Simple Modularity

The log-likelihood for PPM can be easily written:

$$\log L'_{PPM}(C, G, p_{in}, p_{out}) = m_{in}\log p_{in} + m_{out}\log p_{out} +$$
$$(P_{in} - m_{in})\log(1 - p_{in}) + (P_{out} - m_{out})\log(1 - p_{out})\,.$$

Although it can be optimized directly using the algorithms discussed further in the paper, in order to demonstrate a connection of the log-likelihood to simple modularity defined in (1) and to deal with all models in a unified way, we use the conventional trick and say that the number of edges between i and j follows a Poisson distribution with parameter p_{in} or p_{out}. Then we get:

$$\log L_{PPM}(C, G, p_{in}, p_{out}) =$$
$$m_{in}\log p_{in} + m_{out}\log p_{out} - P_{in}p_{in} - P_{out}p_{out}\,. \quad (3)$$

Note that the values p_{in} and p_{out} maximizing (3) are

$$p_{in} = \frac{m_{in}}{P_{in}}, \quad p_{out} = \frac{m_{out}}{P_{out}}\,. \quad (4)$$

In a resent paper by Newman [36] it was shown that if we assume p_{in} and p_{out} to be fixed, then maximizing (3) is equivalent to maximizing simple modularity (1) with

$$\gamma = \frac{P(p_{in} - p_{out})}{m(\log p_{in} - \log p_{out})}\,. \quad (5)$$

4.4 Likelihood for DCPPM and Modularity

The log-likelihood for DCPPM is [36]:

$$\log L_{DCPPM}(C, G, p_{in}, p_{out})$$
$$= m_{in}(\log p_{in} - \log p_{out}) - \frac{p_{in} - p_{out}}{4m} \sum_{C \in \mathcal{C}} D(C)^2$$
$$+ \sum_i d(i) \log d(i) + m \log p_{out} - m p_{out} - m \log(2m) . \quad (6)$$

The values of parameters maximizing this likelihood are:

$$p_{in} = \frac{4\,m\,m_{in}}{\sum_C D(C)^2}, \quad p_{out} = \frac{4\,m\,m_{out}}{4m^2 - \sum_C D(C)^2} . \quad (7)$$

As shown in [36], if p_{in} and p_{out} are fixed, then maximizing $\log L_{DCPPM}(C, G, p_{in}, p_{out})$ is exactly equivalent to maximizing modularity (2) with

$$\gamma = \frac{p_{in} - p_{out}}{\log p_{in} - \log p_{out}} . \quad (8)$$

4.5 Likelihood for ILFR

Let us compute the likelihood for the proposed ILFR model:

$$\log L_{ILFR}(C, G, \mu) = -\frac{1}{2} \sum_{i,j:c(i)=c(j)} d(i)d(j) \left(\frac{(1-\mu)}{D(C_{c(i)})} + \frac{\mu}{2m} \right)$$
$$- \sum_{\substack{i,j:i<j, \\ c(i) \neq c(j)}} \frac{\mu d(i)d(j)}{2m} + \sum_{\substack{(i,j) \in E(G), \\ i<j, c(i)=c(j)}} \log \left(d(i)d(j) \left(\frac{(1-\mu)}{D(C_{c(i)})} + \frac{\mu}{2m} \right) \right)$$
$$+ \sum_{\substack{(i,j) \in E(G), \\ i<j, c(i) \neq c(j)}} \log \frac{\mu d(i)d(j)}{2m} = \sum_{C \in \mathcal{C}} \frac{D_{in}(C)}{2} \log \left(\frac{(1-\mu)}{D(C)} + \frac{\mu}{2m} \right)$$
$$+ m_{out} \log \mu + \sum_i d(i) \log d(i) - m_{out} \log 2m - m , \quad (9)$$

where $D_{in}(C) = \sum_{i \in C} d_{in}(i)$ is twice the number of edges induced by C. Note that the optimal value of μ is hard to find analytically, but it can be approximated numerically by optimizing (9).

The following approximation helps to speed up the optimization algorithms and to make formulas more concise:

$$\log \left(\frac{(1-\mu)}{D(C)} + \frac{\mu}{2m} \right) = \log \frac{(1-\mu)}{D(C)} + \log \left(1 + \frac{\mu D(C)}{(1-\mu)\,2\,m} \right)$$
$$\approx \log \frac{(1-\mu)}{D(C)} + \frac{\mu D(C)}{(1-\mu)\,2\,m} \approx \log \frac{(1-\mu)}{D(C)} ,$$

since $\left| \log \frac{1-\mu}{D(C)} \right| \gg \frac{\mu D(C)}{(1-\mu)\,2\,m}$. This leads to another quality function, which we further refer to as ILFRS (S stands for "simplified"):

$$\log L_{ILFRS}(C, G, \mu) = m_{in} \log(1-\mu) + m_{out} \log \mu - m_{out} \log 2m$$
$$- \sum_C \frac{D_{in}(C)}{2} \log D(C) + \sum_i d(i) \log d(i) - m . \quad (10)$$

The optimal μ according to (10) can now be computed analytically:

$$\mu = \frac{m_{out}}{m} . \quad (11)$$

Note that now we can substitute μ in (10) by the optimal value (11) and obtain a non-parametric quality function:

$$\log L_{ILFRS}(C, G) = m_{in} \log \frac{m_{in}}{m} + m_{out} \log \frac{m_{out}}{m} - m$$
$$- m_{out} \log 2m - \sum_C \frac{D_{in}(C)}{2} \log D(C) + \sum_i d(i) \log d(i) . \quad (12)$$

The obtained function is fairly simple and, as we show by further experiments, it can successfully replace the standard modularity function in many applications.

Let us also note that in contrast to ILFR, the likelihood for the standard LFR model cannot be computed. First, LFR is based on the configuration model, which introduces complex dependences between all edges, making the likelihood intractable. Second, and most importantly, the number of iter-community edges for each vertex is deterministic (the fraction of such edges is μ); as a result, for most of graphs and partitions the likelihood is equal to zero.

5 OPTIMIZATION

5.1 Related Work on Modularity Optimization

First, let us discuss some optimization approaches used in community detection. The most widely used measure to be optimized is modularity (2). The following types of approaches are known.

Greedy optimization. Newman and Girvan [40] proposed a greedy algorithm for modularity optimization, where the clusters are iteratively merged by choosing the pair which leads to a greater modularity increase (while it is positive). A speedup of the greedy algorithm was proposed by Clauset et al. [13]. Some other modifications were also suggested in, e.g., [49, 56]. Probably the most well-known and widely used greedy algorithm is called Louvain [8]. At the beginning each vertex forms its own cluster. Then we create the first level of the partition by iterating through all vertices: for each vertex i we compute the gain in modularity coming from putting i to the community of its neighbor and pick the community with the largest gain, as long as it is positive. After that the first level is formed and we replace the obtained communities by supervertices, and two supervertices are connected by a (weighted) edge if there is at least one edge between vertices of the corresponding communities. Then the process is repeated with the supervertices, and so on, as long as modularity increases. This algorithm was shown to be fast and provide partitions of good quality. We choose Louvain algorithm as the basis for our experiments, the detailed description of its application to our problem is given in Section 5.2.

Simulated annealing. Simulated annealing was first applied to modularity optimization in [23]. This method combines two types of moves: local moves, where a single vertex is shifted from one cluster to another; global moves, consisting of merges and splits of communities. Methods based on simulated annealing are relatively slow and cannot be applied to large datasets.

Spectral optimization. Spectral methods are quite popular in modularity optimization. Such methods use the eigenvalues and eigenvectors of a so-called modularity matrix. Some examples can be found in [37, 53].

Many other algorithms exist, among them are methods based on extremal optimization [18], mathematical programming [2], mean field annealing [30], genetic algorithms [54], etc.

5.2 Proposed Likelihood Optimization Methods

In Section 4, we presented four quality functions based on the likelihood: PPM (3), DCPPM (6), ILFR (9), and ILFRS (10). All functions are parametric: PPM and DCPPM have parameters p_{in} and p_{out}, ILFR and ILFRS have one parameter μ. In this section, we discuss the possible ways to maximize these likelihoods. Below we assume that we are given an arbitrary algorithm $FindPartition(\bar{Q}, G)$ which is able to find a partition by maximizing some quality function $\bar{Q}(C, G)$.

Although the optimization strategies proposed below are able to work with any algorithm $FindPartition$, let us first discuss our particular choice. As we already mentioned, we choose Louvain algorithm [8], which is arguably the most widely used method for community detection. Louvain is fast and allows to process large datasets. Importantly, it can be adapted to all quality functions discussed in this paper. Initially the method is designed to optimize modularity, so it can be directly applied to the likelihood of DCPPM due to their equivalence. To adapt Louvain to other likelihoods, we have to change one part of the original algorithm: when we try to remove a vertex (or supervertex) from one community an add to another, we need to efficiently compute the difference in likelihood. This can be done for any of the considered quality functions: for ILFR and ILFRS the differences can be computed using (9) and (10); for PPM we have to additionally store the sizes of supervertices in order to compute the difference for P_{in} in (1). The rest of the Louvain algorithm remains the same.[4] Therefore, the asymptotic time complexity of our Louvain-based $FindPartition$ and of the original Louvain are the same, up to a constant multiplier (the complexity of Louvain is empirically evaluated to be $O(n \log n)$).

Note that both the original Louvain and our modifications can be applied to the likelihoods with fixed values of parameters p_{in}, p_{out} or μ. Therefore, the key question is how to find the optimal values of these parameters.

Iterative strategy. The simplified version of this strategy was initially proposed in [36] for PPM and DCPPM models. Detailed description of our method is presented in Algorithm 1. Initially, we fix $\gamma = 1$ or $\mu = 0.5$ depending on the null model. Then, we apply *FindPartition* to the corresponding log-likelihood function. Using the obtained partition we can re-estimate the parameters and continue this procedure until convergence or until some maximum allowed number of steps is reached. In our experiments we noticed that for any null model the parameters may end up cyclically varying near some value. Therefore, at each iteration we additionally check if we obtain an already seen parameter and stop in this case. This additional trick allows to significantly reduce the overall number of iterations. In our experiments we observed that we never need more than 50 iterations and the mean value is much smaller, see Section 6 for more details.

Maximization strategy. An alternative strategy is to directly search for parameters which maximize likelihood of the obtained

[4]To make our results reproducible, we made the source code publicly available at https://github.com/altsoph/community_loglike.

Algorithm 1: Iterative strategy

input : graph G, $Model$, algorithm $FindPartition$, N, ε

Initialize $param = 1$ (γ for PPM, DCPPM) or $param = 0.5$ (μ for ILFR, ILFRS), $Params = \emptyset$;

for $i \leftarrow 1$ **to** N **do**
    Define $\bar{Q}$ according to (1) for PPM, (2) for DCPPM, (9) for ILFR, (10) for ILFRS with parameter $param$;
    $C = FindPartition(\bar{Q}, G)$;
    if $Model$ is PPM or DCPPM **then**
        Compute p_{in} and p_{out} according to (4) or (7);
        Compute $param_{new}$ according to (5) or (8);
    if $Model$ is ILFRS **then**
        Compute $param_{new}$ according to (11);
    if $Model$ is ILFR **then**
        Compute $param_{new}$ by optimizing (9);
    if $|param - param_{new}| < \varepsilon$ or $param_{new} \in Params$ **then**
        break;
    Add $param_{new}$ to $Params$;
    $param \leftarrow param_{new}$;
return C

---

partition. Here we can use any black-box optimization method. At each iteration of optimization, for some value of γ or μ, we run *FindPartition* to obtain a partition C, then, using C, we find the optimal parameters according to (4), (7), (11) or by optimizing (9). Finally, we compute the likelihood to be maximized using (3), (6), (9), or (10). In our experiments, for simplicity and reproducibility, we use grid search in this strategy. However, any other method of black-box parameter optimization can be used instead, e.g., random search [5], Bayesian optimization [50], Gaussian processes [22], sequential model-based optimization [6, 27], and so on.

To summarize, the maximization strategy directly maximizes the obtained likelihood, while the iterative one searches for a "stable" parameter, i.e., the one which does not change after applying *FindPartition*. Note that the convergence of the iterative strategy is not guaranteed for real-world networks. Therefore, initially we expected the maximization strategy to be more stable. However, our experiments show that these two strategies demonstrate similar performance, but the iterative one is faster.

Let us mention another possible optimization strategy, which we do not use in our experiments. Given a partition C, for each likelihood function except ILFR, we can compute the optimal values of parameters according to (4), (7), and (11). Therefore, for the corresponding models we can substitute these parameters and obtain non-parametric likelihoods, as we did in (12) for ILFRS. All obtained non-parametric quality functions can potentially be optimized directly since they do not have free parameters. However, we do not consider such strategy since: 1) it cannot be applied to ILFR as there is no analytical formula for optimal μ, 2) this strategy cannot be easily combined with the Louvain algorithm. The reason is that Louvain has several partition levels and partitions obtained on earlier levels cannot be changed later. At the beginning of the algorithm our estimates of μ, p_{in}, and p_{out} are far from optimal:

all vertices form their own communities, so $\mu = 1$, $p_{out} = 1$, and $p_{in} = 0$. As a result, the first level of the partition, which has a big impact on the final quality, is constructed based on non-optimal parameters. In fact, in most cases the algorithm does not even start optimization due to the huge overestimate of μ or p_{out}.

6 EXPERIMENTS

In this section, we conduct an extensive experimental study to compare the discussed models. We start with some preliminaries and discuss evaluation metrics and datasets. In Section 6.3, we measure how well the models describe various real-world networks. Next, we compare all proposed community detection algorithms. Finally, Section 6.5 presents some negative results on the limits of statistical inference algorithms when applied to real-world networks.

6.1 Evaluation Metrics

In order to evaluate the performance of any algorithm, we have to compare the partition C obtained by this algorithm with the ground truth partition C_{GT}. The problem of choosing a good similarity measure for this comparison does not have any standard solution in community detection literature. Different similarity measures can possibly give preferences to different algorithms. Usually, a suitable measure is chosen according to a practical problem at hand. That is why in this work we compute and compare several standard similarity measures. In particular, we consider Normalized Mutual Information (NMI) of two partitions C and C_{GT}, which is often used for the comparison of community detection algorithms [4, 19]. The idea behind NMI is that if two partitions are similar, one needs very little information to infer C_{GT} given C. We also use two well-known similarity measures based on counting correctly and incorrectly classified pairs of vertices: Rand and Jaccard indices [19, 46].

6.2 Datasets

6.2.1 Synthetic networks. We use the LFR model described in Section 2.3 to generate synthetic networks. The parameters of the generated graphs are the following: the number of vertices $n = 10^4$; the parameter of the power-law degree distribution $\gamma = 2.5$; the average degree $\bar{d} = 30$; the parameter of the community size distribution $\beta = 1.5$, with the minimum cluster size 50 and the maximum 600; the mixing parameter $\hat{\mu}$ is varied in range $[0, 1]$.[5] Note that there are other benchmark models proposed in the literature. However, community detection in synthetic networks is not the main focus in this paper, therefore we do not consider such models.

6.2.2 Real-world networks. We collected several networks with different structural properties (see Table 2). In addition to widely used networks, such as Zachary's karate club, dolphin social network, and American college football, we also used annotated books about politics[6] and political blogs [1]. The dataset email-Eu-core is obtained from SNAP,[7] here the ground truth communities correspond to the departments of a research institute. In Cora citation dataset communities correspond to subjects of research papers, while in

Table 2: Real-world datasets; μ, γ_0, and γ_1 are computed for ground truth partitions according to (11), (4, 5) and (7, 8)

Dataset	n	m	k	μ	γ_1	γ_0
Karate club [58]	34	78	2	0.128	0.78	0.78
Dolphins [32]	62	159	2	0.038	0.54	0.55
Football [40]	115	613	11	0.325	2.39	2.57
Political books [38]	105	441	3	0.159	0.86	0.89
Political blogs [1]	1224	16715	2	0.094	0.72	0.72
email-Eu-core [31]	986	16064	42	0.664	2.74	2.80
Cora citation [52]	24166	89157	70	0.458	5.46	6.21
AS [9]	23752	58416	176	0.561	1.15	1.40

Table 3: Log-likelihoods: datasets with ground truth

Dataset	$\log L_{PPM}$	$\log L_{DCPPM}$	$\log L_{ILFR}$
Karate	-206.12	-168.65	**-168.63**
Dolphins	-483.50	-439.52	**-428.64**
Football	**-1384.1**	-1426.7	-1428.4
Political books	-1363.8	**-1235.0**	-1243.3
Political blogs	-73912	-50756	**-50750**
Eu-core	-65559	-48783	**-48483**
Cora	-678306	-593358	**-584730**
AS	-542952	-351537	**-329784**

AS [9] the vertices are annotated with their countries. These are all publicly available datasets we found for non-overlapping community detection.[8]

6.3 Comparison of Likelihoods

Since the main aim of this paper is to analyze and compare several null models having community structure, we start with the following research question: *Which model gives the best fit for real-world complex networks*? To answer this question, we compare the probabilities that real-world networks were generated by each of the null models. Namely, we compared the log-likelihoods (3) for PPM, (6) for DCPPM and (9) for ILFR.[9] Note that (10) is a simplified expression for (9), so we consider it only as a quality function but not as log-likelihood.

In the first experiment, we took the datasets described in Section 6.2.2 and assumed that the partitions are defined by the ground truth cluster assignments provided for these datasets. Then, we computed the optimal parameters p_{in}, p_{out} or μ and used them to compute the corresponding log-likelihood (see Table 3). One can see that PPM is the best model describing the Football dataset, DCPPM is the best for Pol-books, while for all other datasets ILFR has the largest likelihood. Note that ILFR has only one parameter to be tuned while DCPPM has two. Therefore, we initially expected that for many datasets DCPPM may have a larger likelihood. In this case, to decide which model better describes the data, we would have to adapt some information criterion (e.g., the Bayesian information

---

[5]We additionally experimented with synthetic graphs of different edge densities obtained by varying the average degree. The results are omitted since the only difference observed is that the qualities of all algorithms are usually larger for denser graphs.
[6]V. Krebs, unpublished, http://www.orgnet.com/
[7]http://snap.stanford.edu/data/email-Eu-core.html

[8]It is easier to find datasets with ground truth overlapping communities, see, e.g., http://snap.stanford.edu/data/index.html#communities
[9]We intentionally do not consider synthetic networks in this experiment: likelihoods are expected to be heavily affected by the particular synthetic model used. For example, LFR benchmark could give a preference to ILFR null model.

Table 4: Log-likelihoods: datasets without ground truth

Dataset	n	$\log L_{PPM}$	$\log L_{DCPPM}$	$\log L_{ILFR}$
Karate	34	-191.182	-163.990	**-160.154**
Dolphins	62	-417.240	-398.718	**-394.538**
Football	115	**-1364.38**	-1407.43	-1406.79
Political books	105	-1182.90	-1135.57	**-1088.77**
Political blogs	1224	-60653.8	-49912.3	**-49702.3**
Eu-core	986	-57421.1	-46020.9	**-45469.2**
Cora	24K	-512556	-450154	**-425463**
AS	24K	-449088	-244745	**-227917**
Ca-GrQc	5242	-50258.2	-53393.9	**-42259.3**
Ego-Facebook	4039	-241708	-234311	**-207910**
p2p-Gnutella09	8114	-167995	-148594	**-144711**
Wiki-vote	7115	-504395	**-388044**	-388380
Email-Enron	37K	-1192893	-862315	**-803161**
Soc-Epinions1	76K	-2825138	-2087117	**-2000906**
Soc-Slashdot0811	77K	-4011399	-3120963	**-2949833**
ego-Twitter	81K	-6979664	-5781209	**-5314782**

criterion) to our problem, which is a nontrivial task for the models under consideration. Surprisingly, for most of the datasets ILFR has a larger likelihood, which clearly indicates that this model is more suitable for describing real-world networks.[10]

However, as we further discuss in Section 6.4.2, even though for each dataset under consideration we have a ground truth partition (a.k.a. attributes or metadata), this partition can be not an ideal division of vertices into communities and also can be not the only ground truth partition possible. Fortunately, likelihoods can be compared even for datasets without any knowledge about community structure, which allows to compare null models on a much larger variety of datasets. In order to measure the likelihood for any graph, we first have to find a partition that maximizes this value (among all possible partitions), which is feasible only for very small datasets. However, the optimal partition (and the corresponding likelihood) can be approximately found by applying the corresponding maximization algorithm: PPM-max, DCPPM-max or ILFR-max. We performed such comparison on the datasets introduced in Section 6.2.2 as well as on 8 new datasets of various nature downloaded from http://snap.stanford.edu/data/index.html (see Table 4). According to this experiment, for almost all datasets the largest likelihood is again obtained for ILFR model, despite it has fewer free parameters, which means that this model is the best one for describing real-world datasets, which is the main empirical result supporting the introduction of ILFR and the corresponding log-likelihood quality function.

Let us note that the log-likelihoods presented in Table 4 can be underestimated, since we cannot guarantee that our optimization algorithms find exactly the maximum likelihood partition. However, we believe that this does not introduce any bias into this experiment (i.e., does not change the conclusions) since the same optimization procedure is applied for all models. Also, note that in all cases the log-likelihoods in Table 3 are smaller than the corresponding ones in Table 4, which means that our optimization algorithms found

---

[10]Note that using these results we cannot compare ILFR and DCPPM with PPM since ILFR and DCPPM are based on the observed degree sequences, while PPM only uses the cluster assignments.
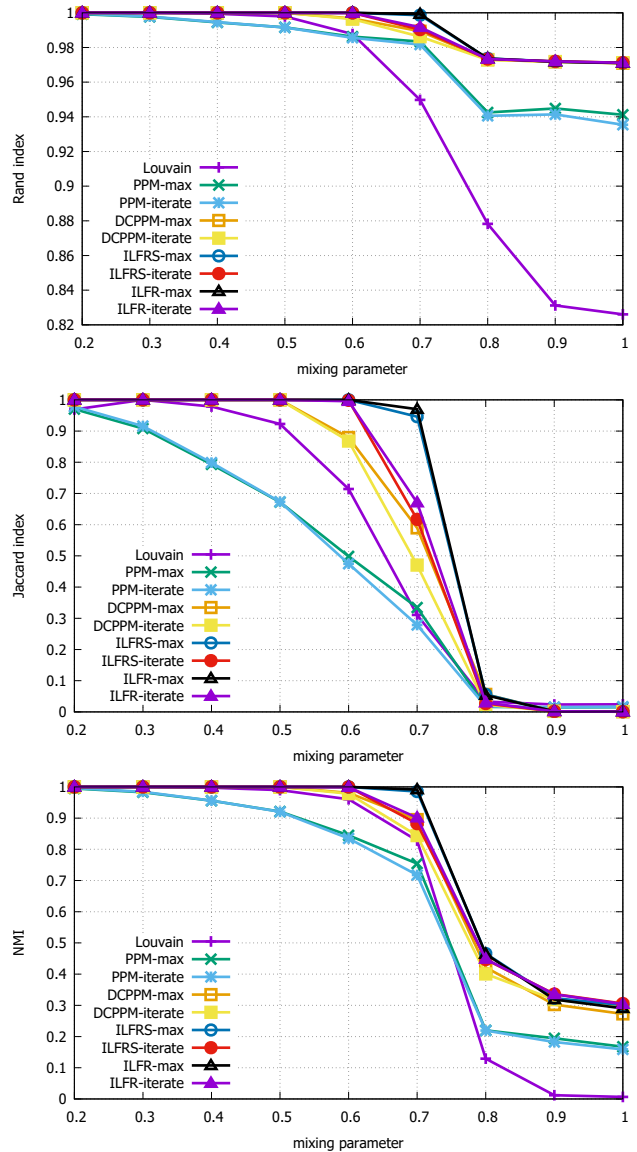

Figure 1: Comparison on synthetic networks

partitions providing better likelihoods for the corresponding models than the ground truth ones, which is expected (since partitions were tuned).

6.4 Community Detection Algorithms

In this section, we compare all algorithms proposed in Section 5.2. Note that we intentionally use only methods based on the likelihood optimization since this paper focuses on the analysis and comparison of null models.

6.4.1 Synthetic networks. First, we compared all algorithms on synthetic networks described in Section 6.2.1. For each $\hat{\mu}$ we generated 5 random samples of LFR and averaged the results. Figure 1 shows the results obtained for all similarity measures under consideration

Table 5: Maximization strategies (Rand index / Jaccard index / NMI)

Dataset	Louvain	PPM	DCPPM	ILFR	ILFRS
Karate	0.732 / 0.470 / 0.586	0.707 / 0.397 / 0.585	**0.766 / 0.523 / 0.667**	**0.774 / 0.535 / 0.687**	**0.774 / 0.535 / 0.687**
Dolphins	**0.634 / 0.351 / 0.499**	0.529 / 0.153 / 0.386	0.600 / 0.286 / 0.483	0.594 / 0.274 / 0.472	0.584 / 0.256 / 0.454
Football	0.970 / 0.720 / 0.922	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**
Pol-books	0.831 / 0.616 / 0.545	0.722 / 0.331 / 0.449	0.817 / 0.583 / 0.536	0.719 / 0.347 / 0.437	0.772 / 0.477 / 0.490
Pol-blogs	0.886 / 0.788 / 0.641	0.574 / 0.153 / 0.234	**0.904 / 0.823 / 0.705**	0.860 / 0.738 / 0.570	0.879 / 0.778 / 0.625
Eu-core	0.869 / 0.218 / 0.584	0.942 / 0.207 / 0.661	**0.964 / 0.434 / 0.740**	0.953 / 0.371 / 0.727	0.960 / 0.410 / 0.739
Cora	0.943 / **0.127** / 0.460	**0.978** / 0.044 / 0.547	0.978 / 0.050 / 0.533	0.978 / 0.063 / **0.551**	0.978 / 0.060 / **0.551**
AS	0.821 / **0.198** / **0.489**	0.826 / 0.009 / 0.422	**0.826** / 0.026 / 0.461	0.826 / 0.018 / 0.438	0.826 / 0.018 / 0.438

Table 6: Iterative strategies (Rand index / Jaccard index / NMI)

Dataset	Louvain	PPM	DCPPM	ILFR	ILFRS
Karate	0.732 / 0.470 / 0.586	0.743 / 0.480 / 0.612	0.750 / 0.502 / 0.612	0.766 / 0.523 / 0.667	0.760 / 0.519 / 0.633
Dolphins	**0.634 / 0.351 /** 0.499	0.548 / 0.190 / 0.403	0.600 / 0.286 / 0.481	0.576 / 0.238 / 0.449	0.575 / 0.238 / 0.448
Football	0.970 / 0.720 / 0.922	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**	**0.993 / 0.916 / 0.972**
Pol-books	**0.831 / 0.616 / 0.545**	0.740 / 0.377 / 0.464	**0.843 / 0.651 / 0.554**	0.760 / 0.445 / 0.481	0.770 / 0.475 / 0.489
Pol-blogs	0.886 / 0.788 / 0.641	0.582 / 0.171 / 0.240	**0.902 / 0.820 / 0.694**	0.862 / 0.743 / 0.576	0.877 / 0.773 / 0.618
Eu-core	0.869 / 0.218 / 0.584	0.939 / 0.205 / 0.656	**0.965 / 0.439 / 0.742**	0.948 / 0.351 / 0.713	0.955 / 0.387 / 0.725
Cora	0.943 / **0.127** / 0.460	**0.978** / 0.049 / 0.547	0.978 / 0.052 / 0.534	0.978 / 0.058 / **0.549**	0.978 / 0.061 / **0.550**
AS	0.821 / **0.198** / **0.489**	0.826 / 0.011 / 0.426	**0.826** / 0.025 / 0.460	0.826 / 0.017 / 0.436	0.826 / 0.017 / 0.435

(Rand, Jaccard and NMI). Note that in most cases these similarity measures are consistent, i.e., for a given mixing parameter they rank the algorithms similarly. However, Jaccard index is less descriptive for large mixing parameters.

Based on the obtained results, several important observations can be made. For all values of $\hat{\mu}$, the best (or very close to the best) quality is obtained by ILFRS-max and ILFR-max strategies. For both ILFR and ILFRS it turns out that maximizing the likelihood is better than iteratively estimating the parameters, which supports our intuition that better partitions are expected to have higher likelihoods. For small values of $\hat{\mu}$ the worst results are obtained by PPM (both strategies), while for large values Louvain algorithm is the worst.

We also measured statistical significance of the obtained improvements: for each $\hat{\mu}$, we applied the algorithms to 5 random samples of LFR and used the paired t-test. For instance, it turns out that according to NMI ILFRS-max never looses significantly and it is significantly better (p-value < 0.05) than: DCPPM-max for $\hat{\mu} \in \{0.7, 0.9, 1\}$; DCPPM-iterate for $\hat{\mu} \in \{0.7, 0.8\}$; PPM-max and PPM-iterate for all $\hat{\mu} \in [0.2, 1]$; ILFR-iterate and ILFRS-iterate for $\hat{\mu} = 0.7$; and Louvain for $\hat{\mu} \in [0.4, 1]$. These results additionally support the choice of the quality function L_{ILFRS} (12) for community detection instead of the standard modularity function.

6.4.2 Real-world complex networks. It is argued in several papers that ground truth community labels (a.k.a. metadata) available for some real-world networks should not be used to analyze and compare community detection algorithms [43]. Indeed, such labels are usually obtained using some discrete-valued vertex attributes which are not guaranteed to be reasonable community assignments. For example, users in a social network can be split by gender, city, or

interests, and each of these partitions could be treated as ground truth cluster assignments. An algorithm performing better for "city" labels can be worse for "interests" ones. As a result, no meaningful conclusion can be made based on such comparison. However, for the sake of completeness, we compare all algorithms on real-world networks, since it is still a standard practice in community detection literature.

Tables 5 and 6 present the results for the iterative and maximization strategies, respectively (in both cases we add Louvain for comparison).[11] Note that we cannot properly measure statistical significance since we cannot sample several copies for each dataset. However, we can account for the randomness included in the algorithms: they all order vertices at each iteration randomly. To do this, for each dataset we run all algorithms 5 times and then apply the unpaired t-test to compare them. We put numbers in bold in Tables 5 or 6 if there is a group of algorithms without significant differences (p-value > 0.05) inside the group and with significant differences with the rest of the algorithms.

The results in general are not consistent: although DCPPM is often the best, the winning model may depend on a graph under consideration and on a target metric. For example, Louvain wins on Cora and AS according to Jaccard index, but it is the worst on the same datasets according to Rand index. This supports the claim that ground truth community assignments can be noisy or irrelevant.

We also noticed that iterative and maximization strategies usually have similar performance and the choice between them is not straightforward. For example, for ILFRS it is often better to apply maximization strategy, while for PPM iterative one should be

---

[11]Recall that Louvain optimizes modularity, so it is similar to DCPPM but with fixed values of p_{in} and p_{out}.

Table 7: Quality functions: ground truth vs optimization

Dataset	Modularity		$\log L_{PPM}$		$\log L_{DCPPM}$		$\log L_{ILFRS}$		$\log L_{ILFR}$	
	GT	Louvain	GT	PPM-max	GT	DCPPM-max	GT	ILFRS-max	GT	ILFR-max
Karate	0.3715	0.4188	-206	-191	-169	-164	-176	-168	-169	-160
Dolphins	0.3735	0.5233	-483	-417	-439	-399	-434	-405	-429	-395
Football	0.5877	0.6043	-1384	-1364	-1427	-1407	-1447	-1425	-1428	-1407
Pol-books	0.4149	0.5205	-1364	-1183	-1235	-1136	-1285	-1125	-1243	-1089
Pol-blogs	0.4053	0.4267	-73913	-60654	-50756	-49912	-51843	-50614	-50750	-49702
Eu-core	0.3138	0.4211	-65559	-57421	-48783	-46021	-49073	-46202	-48483	-45469
Cora	0.5165	0.7875	-678306	-512556	-593358	-450154	-585837	-425657	-584730	-425463
AS	0.1708	0.6307	-542952	-449088	-351537	-244745	-338397	-227192	-329784	-227917

preferred. Based on this observation, we propose using the iterative strategy, which is faster (it converges after a small number of iteration, as discussed below).

The performance of ILFRS is in general slightly better than of ILFR. Taking into account our results on synthetic networks, we propose using the faster ILFRS instead of ILFR in all practical applications.

Finally, we analyzed the speed of convergence for iterative strategies on real-world datasets. Namely, we measured the number of iterations made by Algorithm 1 before some stopping criteria is reached. Note that the algorithm never stopped because of reaching the maximum allowed number of iterations $N = 50$, so we either observed a convergence or applied the cycling criteria (i.e., encountered an already seed parameter). The average number of iterations obtained for PPM is 12.6 (in all cases the algorithm converged). For DCPPM we got 6.2 (converged in all cases except AS, where we applied cycling stopping criteria after 18 iterations). For ILFRS we have 5.4 iterations (cycling criteria is applied to 3 datasets). For ILFR we got 4.4 (again, with 3 applications of cycling criteria).

6.5 Statistical Inference, Negative Result

Having noticed unstable results for real-world datasets, we tried to answer the following research question: *Is any of the null models suitable for community detection in real-world graphs?*

In order to answer this question, for each quality function we compare its value for the ground truth partition with its value for the partition obtained by the corresponding maximization algorithm (see Table 7). For all quality functions, including the widely used modularity, and for all datasets the ground truth partition has a lower value of the quality function. This means that further optimization of *any* quality function would not lead us towards the ground truth partition, which is a negative observation. In particular, there is no hope in improving the results obtained by our algorithms by replacing Louvain-based *FindPartition* with some more effective maximization algorithm.

Note that in the literature it is often assumed that a partition with larger modularity is better and, as a result, an algorithm which leads to a partition with larger modularity is better. However, our observation above demonstrates that on the considered real-world datasets it is not the case. We also performed additional experiments and noticed that in almost all cases the value of a quality function for the ground truth partition is lower than for the partition obtained

by *any* discussed optimization algorithm (not necessary optimizing the same quality function), which is an even stronger negative observation.

The following conclusion can be made: either the ground truth metadata contained in the considered real-world networks is not a good descriptor for a community structure or statistical inference algorithms based on all null models discussed in this paper are unable to detect real-world communities.

7 CONCLUSION

In this paper, we focused on the comparison of null models which can be used by likelihood optimization algorithms for community detection. We compared two well-known models, PPM and DCPPM, and a new model, ILFR, which has only one parameter and is proved to preserve the desired expected degree sequence. For the new model we have written the log-likelihood, both in parametric and self-contained forms. To maximize the parametric log-likelihood functions, we proposed and compared two optimization strategies: maximization and iterative.

The most important conclusion is that the proposed model, ILFR, is the best one for describing most of the considered real-world complex networks according to the likelihood of the observed graph structures, despite the fact that it has only one free parameter. We argue that the likelihood can be considered as the main argument in evaluating the null models instead of the direct comparison of community detection algorithms. The reason is that one cannot fully rely on ground truth cluster labels available for real-world networks. Also, we demonstrated that ILFR-based algorithms have the best performance on synthetic networks. Based on the obtained results, we believe that the proposed ILFR-based quality function (12) can be successfully used as a target for optimization algorithms, instead of the widely adopted modularity.

A natural direction for the future research is to analyze null models for overlapping community detection. This would be useful since many observed networks have overlapping communities. However, fundamental analysis of this problem is complicated by the fact that null models with overlapping communities are less developed and more complex for mathematical analysis.

REFERENCES

[1] Lada A Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*. ACM, 36–43.

[2] Gaurav Agarwal and David Kempe. 2008. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B* 66, 3 (2008), 409–418.

[3] Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. 2014. Learning latent block structure in weighted networks. *Journal of Complex Networks* 3, 2 (2014), 221–248.

[4] James P Bagrow. 2008. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 05 (2008), P05001.

[5] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13, Feb (2012), 281–305.

[6] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*. 2546–2554.

[7] Peter J Bickel and Aiyou Chen. 2009. A nonparametric view of network models and Newman–Girvan and other modularities. *Proceedings of the National Academy of Sciences* 106, 50 (2009), 21068–21073.

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[9] Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. 2010. Sustaining the internet with hyperbolic mapping. *Nature communications* 1 (2010), 62.

[10] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. 2017. Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)* 50, 4 (2017), 54.

[11] Yi Chen, XL Wang, Xin Xiang, BZ Tang, and JZ Bu. 2015. Network structure exploration via Bayesian nonparametric models. *Journal of Statistical Mechanics: Theory and Experiment* 2015, 10 (2015), P10004.

[12] Fan Chung and Linyuan Lu. 2002. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics* 6, 2 (2002), 125–145.

[13] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E* 70, 6 (2004), 066111.

[14] Anne Condon and Richard M Karp. 2001. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms* 18, 2 (2001), 116–140.

[15] Jernej Copic, Matthew O Jackson, and Alan Kirman. 2009. Identifying community structures from network data via maximum likelihood methods. *The BE Journal of Theoretical Economics* 9, 1 (2009).

[16] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4, 5 (2011), 512–546.

[17] J-J Daudin, Franck Picard, and Stéphane Robin. 2008. A mixture model for random graphs. *Statistics and computing* 18, 2 (2008), 173–183.

[18] Jordi Duch and Alex Arenas. 2005. Community detection in complex networks using extremal optimization. *Physical review E* 72, 2 (2005), 027104.

[19] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3 (2010), 75–174.

[20] Santo Fortunato and Marc Barthélemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104, 1 (2007), 36–41.

[21] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *Physics Reports* 659 (2016), 1–44.

[22] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. 2017. Google vizier: A service for black-box optimization. In *International Conference on Knowledge Discovery and Data Mining*. ACM, 1487–1495.

[23] Roger Guimera and Luis A Nunes Amaral. 2005. Functional cartography of complex metabolic networks. *nature* 433(7028), 7028 (2005), 895–900.

[24] Matthew B Hastings. 2006. Community detection as an inference problem. *Physical Review E* 74, 3 (2006), 035102.

[25] Jake M Hofman and Chris H Wiggins. 2008. Bayesian approach to network modularity. *Physical review letters* 100, 25 (2008), 258701.

[26] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks* 5, 2 (1983), 109–137.

[27] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*. Springer, 507–523.

[28] Brian Karrer and Mark EJ Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E* 83, 1 (2011), 016107.

[29] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Physical review E* 78, 4 (2008), 046110.

[30] Sune Lehmann and Lars Kai Hansen. 2007. Deterministic modularity optimization. *The European Physical Journal B-Condensed Matter and Complex Systems* 60, 1 (2007), 83–88.

[31] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 2.

[32] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. 2003. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* 54, 4 (2003), 396–405.

[33] Fragkiskos D Malliaros and Michalis Vazirgiannis. 2013. Clustering and community detection in directed networks: A survey. *Physics Reports* 533, 4 (2013), 95–142.

[34] Aaron McDaid and Neil Hurley. 2010. Detecting highly overlapping communities with model-based overlapping seed expansion. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. IEEE, 112–119.

[35] Michael Molloy and Bruce Reed. 1995. A critical point for random graphs with a given degree sequence. *Random structures & algorithms* 6, 2-3 (1995), 161–180.

[36] MEJ Newman. 2016. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319* (2016).

[37] Mark EJ Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 74, 3 (2006), 036104.

[38] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[39] Mark EJ Newman and Aaron Clauset. 2016. Structure and inference in annotated networks. *Nature communications* 7 (2016).

[40] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

[41] Mark EJ Newman and Elizabeth A Leicht. 2007. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences* 104, 23 (2007), 9564–9569.

[42] Krzysztof Nowicki and Tom A B Snijders. 2001. Estimation and prediction for stochastic blockstructures. *J. Amer. Statist. Assoc.* 96, 455 (2001), 1077–1087.

[43] Leto Peel, Daniel B Larremore, and Aaron Clauset. 2017. The ground truth about metadata and community detection in networks. *Science advances* 3, 5 (2017), e1602548.

[44] Tiago P Peixoto. 2015. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E* 92, 4 (2015), 042807.

[45] José J Ramasco and Muhittin Mungan. 2008. Inversion method for content-based networks. *Physical Review E* 77, 3 (2008), 036122.

[46] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[47] Jörg Reichardt and Stefan Bornholdt. 2006. Statistical mechanics of community detection. *Physical Review E* 74, 1 (2006), 016110.

[48] Martin Rosvall and Carl T Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.

[49] Philipp Schuetz and Amedeo Caflisch. 2008. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E* 77, 4 (2008), 046112.

[50] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. 2015. Scalable bayesian optimization using deep neural networks. In *International Conference on Machine Learning*. 2171–2180.

[51] Myra Spiliopoulou. 2011. Evolution in social networks: A survey. In *Social network data analytics*. Springer, 149–175.

[52] Lovro Šubelj and Marko Bajec. 2013. Model of complex networks based on citation dynamics. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 527–530.

[53] Yudong Sun, Bogdan Danila, K Josić, and Kevin E Bassler. 2009. Improved community structure detection using a modified fine-tuning strategy. *EPL (Europhysics Letters)* 86, 2 (2009), 28004.

[54] Mursel Tasgin, Amac Herdagdelen, and Haluk Bingol. 2007. Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv:0711.0491* (2007).

[55] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[56] Ken Wakita and Toshiyuki Tsurumi. 2007. Finding community structure in mega-scale social networks. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 1275–1276.

[57] Jaewon Yang and Jure Leskovec. 2014. Structure and overlaps of ground-truth communities in networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 26.

[58] Wayne W Zachary. 1977. An information flow model for conflict and fission in small groups. *Journal of anthropological research* 33, 4 (1977), 452–473.

[59] Hugo Zanghi, Christophe Ambroise, and Vincent Miele. 2008. Fast online graph clustering via Erdős–Rényi mixture. *Pattern recognition* 41, 12 (2008), 3592–3599.