Random Graph Generators for Hyperbolic Community Structures

Saskia Metzler[1] and Pauli Miettinen[2]

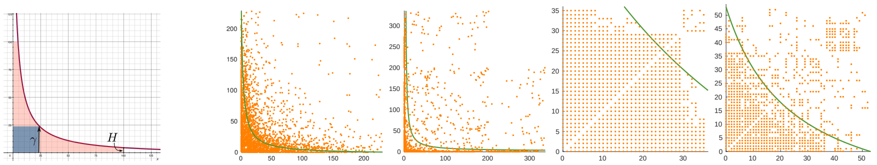
[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany,
smetzler@mpi-inf.mpg.de
[2] University of Eastern Finland, Kuopio, Finland, pauli.miettinen@uef.fi

Abstract. Proper testing of graph mining algorithms, for example, algorithms for community detection, requires the capability of creating realistic random graphs. As our understanding of real graph communities evolves, so should the random graph generators evolve, too. In this work, we propose a random graph generator called HyGen that, unlike the existing random graph generators, is designed to preserve the community structure of real networks, especially the commonly observed hyperbolic intra-community connectivity structure. The generated graphs will also preserve the total degree distributions and clustering coefficients of the original graph without introducing too much determinism. In addition, we also propose realistic distributions for the parameters controlling the hyperbolic shape of the communities.

Keywords: random graphs, graph generators, community detection, hyperbolic community structure

1 Introduction

Real-world networks often not only display a modular composition, but also characteristic structures within the constituents. Previous work has established that structure of communities is described well by a *hyperbolic model* [4, 18]. This model can express the particular core-tail structure which is frequently observed in real-world networks and is suitably general to also represent clique-like connectivity (see Fig. 1b). Especially communities in social networks show a pronounced core-tail structure: a small fraction of the members have strong ties to each other and form the core. The majority of members only have ties to the core and not to each other [2, 19, 21, 23].
Understanding the organization of such networks is a primary goal of social sciences and requires competent algorithms for detecting and describing the structures. The algorithms have to be tested, though, and a thorough testing requires significant amounts of reliably labelled test data – which is often not available. Good random graph generators can be used to alleviate this problem.
There exist a variety of different graph generators: the Watts–Strogatz model [24], the Barabási–Albert model [3], the stochastic block model [10], and so on. They are designed with a focus on different features of (real-world) graphs, such as small-world or scale-free properties. To the best of our knowledge, however, no existing random graph generator is designed to model graphs consisting of hyperbolic communities.



(a) γ and H

(b) Examples of hyperbolic communities from [18]

Fig. 1: Visualization of parameters for the hyperbolic model, and real-world examples. Models are shown on the degree-ordered adjacency matrix of each community.

As we believe that many real-world graphs actually follow a hyperbolic model, we introduce a novel random graph generator to fill this gap. HyGen generates modular networks with realistic intra-community structures using parameter distributions derived from observations on real graphs.

2 Existing Random Graph Generators

The emphasis of our approach is to model the intra-community structure of graphs realistically. Many existing random graph generators do not consider the concept of community structure at all. The classical Erdős–Rényi [8] model is one of them, as are the Barabási–Albert [3] model and the Forest fire model [16]. Approaches for *graph expansion* [22, 25] typically focus on the global level of modelling real-world graphs. Likewise do hyperbolic geometric graphs [13] whose construction is based on *hyperbolic geometry*. Despite the related-sounding name, there is no direct resemblance to our model. Kronecker graphs [15] permit community structure, but due to their recursive construction using the Kronecker product, communities consist of self-similar building blocks and do not vary in size.

Graph generators that permit community structure and that we consider most relevant in comparison to our method are the stochastic block model (SBM) [10], in particular the variation called degree-corrected SBM (DC-SBM) [11], the R-MAT model [6], and the Lancichinetti–Fortunato–Radicchi (LFR) benchmark [14]. Notice that neither DC-SBM, nor R-MAT, nor LFR provide solutions to the modelling task we aim to solve. We tested to what extent an ideal hyperbolic structure could be captured by the different graph generators and found that none of them captures it well (we omit the details). Some reasons for this are discussed below.

The stochastic block model (SBM) is a popular random graph model that partitions vertices into blocks. The vertices within each block are stochastically equivalent. But SBMs cannot model uneven degree distributions within communities. Degree-corrected SBMs (DC-SBM) [11] are a variation where an additional degree parameter is incorporated for each vertex such that uneven edge probabilities can be accounted for. A crucial detail of this approach is that the probability of a node forming an edge is a global property. Degree-generated SBMs [26], although they overcome the shortcoming of DC-SBMs to not separate vertices based on degree even when that would be the correct partitioning, are similar in that respect. We in contrast assume different probabilities for a node to form inter- or intra-community edges.

R-MAT [6] is based on the recursive construction of an adjacency matrix. The proposed algorithm recursively subdivides this matrix into four equally sized partitions and distributes the edges within according to partition-specific probabilities. This special construction limits the shape of the communities that can be attained. The Lancichinetti–Fortunato–Radicchi (LFR) benchmark [14] is a graph generator proposed to test community detection algorithms. It can produce overlapping community structures as well as weighted and directed networks. It extends the Girvan–Newman benchmark [9] with emphasis on features of real-world graphs such as heterogeneous distributions of the overall node degree and the community size. Still we observe non-realistic intra-community structures showing a nearly uniform degree distribution.

3 Hyperbolic Community

Given an undirected graph $G = (V, E)$ with n nodes and m edges, we assign a number from $\{0, \ldots, n-1\}$ to the vertices and use (i, j) to denote both a pair of vertices and the (potential) undirected edge between them. We call a tuple (V_C, π_C, Θ_C) a *community* C. The set $V_C \subseteq V$ contains the nodes of the community, and we write $n_C = |V_C|$. The permutation $\pi_C \colon V_C \to \{0, \ldots, n_C - 1\}$ orders the nodes, and Θ_C denotes a set of parameters. The hyperbolic community model assumes the nodes to be ordered according to their degrees inside the community. In the model, not every edge between the nodes in V_C is necessarily part of the community – otherwise all communities would be cliques.

Following [18], community models are defined using functions $f \colon \{0, \ldots, n_C - 1\} \times \{0, \ldots, n_C - 1\} \times \Theta_C \to \{0, 1\}$ operating on a set of parameters Θ_C and deciding for any pair of vertices $(i, j) \in \{0, \ldots, n_C - 1\} \times \{0, \ldots, n_C - 1\}$ if an edge between i and j is part of the community or not. Notice that the function f only gets the indices relative to the subgraph, not to the full graph. Thus, to test a pair $(i, j) \in V_C \times V_C$, we need to compute $f(\pi_C(i), \pi_C(j), \Theta_C)$.

Metzler et al. [18] define multiple equivalent parameter sets Θ. We describe our model using $\Theta = \texttt{fixed}(\gamma, H)$, which provides an immediate intuition about the shape of the connectivity pattern of the community: γ defines the size of the core (a clique) and H indicates how thick the tail[3] is (see Fig. 1a). In the subsequent analysis it is sometimes useful to consider $\Theta = \texttt{hyperbolic}(p, \theta)$ instead. The parameters then have an immediate geometric interpretation: p defines the centre of an hyperbola at $(-p, -p)$, and an edge (i, j) is in considered to be part of the community if $(i + p)(j + p) \leq \theta$ (see also [18]).

Yet alternatively, the model can equivalently be expressed in terms of $\Theta = \texttt{mixture}(x, \Sigma)$ [18]. With this formulation, the generality of the hyperbolic model is most evident. The mixture parameter $x \in [-1, 1]$ indicates how much the model looks like a line and how much like a hyperbola centered at the origin: $(1 - |x|)(i \cdot j) + x(i + j) \leq \Sigma$. Controlling the boundary condition Σ while fixing $x = 0$ will yield communities that strictly follow a power law connectivity pattern.

[3] More commonly, intra-communities structures are described as *core* and *periphery* [5]. We use the notion of *core* and *tail* instead since the hyperbolic model allows for more shape variations than the term periphery implies: Tails may get progressively thinner while nodes in the periphery are assumed to be evenly connected to the core.

```
Data: distributions $D_{size}$, $D_\gamma$, $D_H$, densities $d_{inside}$, $d_{outside}$, number of communities $k$
Result: random graph $G$
for $i = 1 : k$ do
    draw size $s$ from $D_{size}$, $\gamma$ from $D_\gamma$, and $H$ from $D_H$
    scale $\gamma$ according to $s$, and $H$ according to $\gamma$
    make model fixed($\gamma$, $H$)
    select edges to discard uniformly at random to reach $d_{inside}$
    plant result into $G$
apply noise $d_{outside}$ to the outside community area of $G$
return $G$
```

Algorithm 1: HyGen algorithm

4 Our Model

In this section, we propose HyGen, the random graph generator that accounts for specific intra-community connection patterns that are frequently observed in real-world social networks [4, 18]. We first explain the construction of individual communities and then describe how a graph of multiple such communities is obtained.

To generate a single random hyperbolic community of size n_C, we need to sample its core size γ, and its tail height H from predefined distributions. Based on observations in real-world data sets, we assume that γ (relative to n_C) can be well modelled through a Normal distribution with mean μ and variance σ^2, and H (as a fraction of γ) follows an Exponential distribution with a decay rate of λ. (More on the distribution functions in Section 6.) Assuming μ, σ^2, and λ are given, the resulting community C of size n_C then perfectly follows the model defined by the sampled parameter set `fixed`(γ, H). Real communities however are typically inexact. We apply a uniform noise model with separate parameters d_{inside} and $d_{outside}$ for edges inside and outside the community. This means that every edge from inside C is retained with probability d_{inside}, and with probability $d_{outside}$, edges are introduced outside the community.

With individual hyperbolic communities as building blocks HyGen generates graphs of k communities, obtained sampling their sizes from the distribution of community sizes D_{size} and drawing the parameters γ and H for every community (see Algorithm 1). While our experiments suggest to draw from a Generalized extreme value distribution, a power law function as used in [14] is a considerable alternative.

Notice that we make the following assumptions: Noise is not only constant within each community but also the same among all communities. We model the area outside communities with a uniform density. We assume the size and shape of the communities to be uncorrelated. Communities are assumed to be non-overlapping.

The time complexity of Algorithm 1 is as follows. For a community C, let E_C^p be the edges a “perfect” community (i.e. one with no noise) would have and let E^p be that for the full graph. Let E^n be the set of edges noise *adds* to the graph (inter and intra community). Then, drawing the parameters and adjusting them is $O(1)$ operation, making the model takes $O(n_C)$, and sampling the edges to discard from the community can be done in $O(|E_C^p|)$ [12, p. 137]. Repeated k times this becomes $O\big(k(n_c + |E_C^p|)\big) = O(|V| + |E^p|)$. This leaves the part to add the noise; to that end, we need to do sampling without replacement over a population of

$O(|V|^2 - |E_p|)$ edges, taking essentially linear time. When $|E_p|$ and $|E^n|$ are small compared to $|V|^2$ (i.e. the graph is sparse), we can sample with replacement to obtain practically the same result, taking $O(|E^n|)$ time. In total, the full running time of Algorithm 1 is $O(|V| + |E^p| + |E^n|)$ which is only slightly more than $O(|V| + |E|)$.
Finally, it is worth noticing that our model can also be generalized as a *graphon* [20]. The division to communities works the same way as when modelling the stochastic block models as graphons; only the edge probability inside the communities is not uniform but rather depends on the model for the community. The modelling as a graphon facilitates the analysis of infinitely large random graphs and the convergence and concentration features of our model, but these are beyond the scope of this manuscript.

5 Properties of HyGen Random Graphs

HyGen preserves important measures of network connectivity. The degree distribution and the clustering coefficient are frequently used to describe the connectivity patterns of networks [1]. In this section, we show that these measures are retained when generating a new graph from the parameters observed in an existing network. For a graph G, consisting of disjoint communities C that perfectly follow the hyperbolic models `hyperbolic`(p, θ) we have:

Lemma 1. *The degree distribution* $d\colon V_C \to \{1, \ldots, n_c\}$ *of* C *is determined through the parameters* p *and* θ.

Proof. The model defines that an edge $(i, j) \in \texttt{hyperbolic}(p, \theta)$ if $(i+p)(j+p) \leq \theta$. This inequality can be reformulated such that

$$j \leq \theta/(i+p) - p \; . \tag{1}$$

For every $i \in V_C$, the highest integer j that fulfils (1) is equal to the degree of node i, and hence $d(i) = \max\{j \in N_C : j \leq \theta/(i+p) - p\}$ defines the degree distribution. □

The more intuitive integer parameters H and γ that indicate the size of the core and the tail can alternatively be used in this derivation:

Corollary 1. *The degree distribution of* C *is determined through* H *and* γ.

This follows directly from Eqs. (7) and (8) in [18].
We show that the same also holds for the entire graph G.

Lemma 2. *The degree distribution of* G *is determined by the parameters of its hyperbolic communities.*

Proof. Lemma 1 applies for every community in G. As the communities are disjoint, we obtain the overall degree distribution choosing p and θ according to the respective community and evaluating the inequality for index $\pi_C(i)$ referencing the node within that community (see Section 3),

$$d(i) = \sum_{C \in G} \max\{\pi_C(j) \in V_C : \pi_C(j) \leq \theta_C/(\pi_C(i) + p_C) - p_C\} \; . \quad \square \tag{2}$$

In real-world data sets, the assumption of perfect hyperbolic models we made for the above result is typically not met. Although the hyperbolic model covers a variety of connectivity patterns, such as the classic power law-like structure, as well as the extremes of a star and a clique, real-world data is often noisy and the hyperbolic models only approximately describe the data. We now assess how much the error in the modelling affects the resulting degree distribution.
Suppose $q \in [0,1]$ is the average noise of graph G. That is, a fraction of q edges are missing from inside the communities of G and the outside-community area has a fraction of q surplus edges, i.e. $d_{outside} = q$, $d_{inside} = 1-q$. Then, for node i with a unperturbed degree of $d(i)$, the expected degree $\bar{d}(i)$ is

$$\bar{d}(i) = d(i) - qd(i) + q(n - d(i)) = d(i) + q(n - 2d(i)) \ . \tag{3}$$

The relative degree of a node i is the fraction $\alpha(i)$ of all nodes of G to which i is connected. Hence, the relative expected degree is $\bar{d}(i)/n = \alpha(i) + q(1-2\alpha(i))$. Noise has the strongest effect on a node if the degree $d(i)$ is near its limits, i.e. if $\alpha \approx 1$ or $\alpha \approx 0$. For a node with $\alpha = 0.5$, the presence of noise on expectation will not affect the degree at all. The star pattern is an example of a graph where the degree distribution is heavily impaired when the noise factor q is high.
The clustering coefficient is also determined through the parameters of the hyperbolic models that constitute a graph G. Two well-known variations of the clustering coefficient exist: the global and the local [1]. While the former is the overall ratio of triangles to wedges in a graph, the latter is computed per node and denotes the fraction of triangles around a node.
Suppose C is a perfect hyperbolic community `hyperbolic`(p,θ).

Lemma 3. *The local clustering coefficient CC_i of the nodes of C is determined through p and θ.*

Proof. The local clustering coefficient for node i of C is given by

$$CC_i = \frac{2\left|\left\{(j,h) : j,h \in \mathit{\Gamma}(i), (j,h) \in E\right\}\right|}{d(i)(d(i)-1)} \ , \tag{4}$$

where $\mathit{\Gamma}(i)$ denotes the set of nodes directly connected to i. The nominator counts twice the edges between nodes j and h that are both connected to i. As C perfectly follows `hyperbolic`(p, θ) we may use (1) to express $\mathit{\Gamma}(i)$ as $\mathit{\Gamma}(i) = \left\{h \in V_C : h \le \theta/(i+p) - p\right\}$. The set of edges, E, is $E = \left\{i,j \in V_C : j \le \theta/(i+p) - p\right\}$. For the degree $d(i)$, used in the denominator to indicate the size of the neighbourhood, Lemma 1 applies. □

To make a similar statement about the global clustering coefficient CC, we first need some definitions. Define the indicator function $\chi(i,j)$ as

$$\chi(i,j) = \begin{cases} 1 & j \le \theta/(i+p) - p \\ 0 & \text{otherwise} \ . \end{cases} \tag{5}$$

This function returns for every pair of nodes i and j of community C whether there exists an edge between them according to `hyperbolic`(p, θ).
To count the number of triangles in C, we define $T(i,j,h) \colon \mathbb{N}^3_{\ge 0} \to \{0,1\}$ as

$$T(i,j,h) = \chi(i,j) \cdot \chi(i,h) \cdot \chi(j,h) \ . \tag{6}$$

$T(i,j,h)$ decides on the basis of the hyperbolic model whether a triangle exists between i, j, and h by checking the presence of the three possible edges.
For the existence of wedges in C, we define $W(i,j,h)\colon \mathbb{N}^3_{\geq 0} \to \{0,1\}$ as

$$W(i,j,h) = (1-\chi(i,j))\cdot\chi(i,h)\cdot\chi(j,h) + \chi(i,j)\cdot(1-\chi(i,h))\cdot\chi(j,h) + \chi(i,j)\cdot\chi(i,h)\cdot(1-\chi(j,h))\,. \quad (7)$$

This function checks the presence of exactly two out of the three possible edges between i, j, and h within C. With these functions defined, we now show the dependency between the clustering coefficient and the hyperbolic model.

Lemma 4. *The global clustering coefficient CC of a community C is determined through p and θ.*

Proof. The global clustering coefficient of a graph is three times the number of triangles divided by the number of wedges. If C perfectly follows the hyperbolic model `hyperbolic`(p,θ), we can use (6) to check for the presence of a triangle for every node triple (i,j,h) and (7) to check for the presence of a wedge. Thus, we compute

$$CC = \frac{3\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1}\sum_{h=j+1}^{n-1} T(i,j,h)}{\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1}\sum_{h=j+1}^{n-1} W(i,j,h)}\,. \quad \square \quad (8)$$

We could compute the number of triangles with only a single sweep over the nodes: there are $\binom{\gamma+1}{3}$ triangles in the core, and every i with $d(i) \geq 2$ adds $\binom{d(i)}{2}$ more triangles because it only has connections to the core. To the best of our knowledge, there is no similar expression to determine the number of wedges.

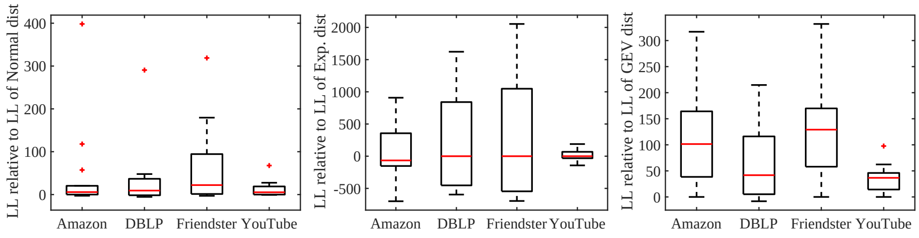
Lemma 5. *The local and global clustering coefficient of G are determined through the parameters of the hyperbolic communities of G.*

Proof. For a graph G consisting of multiple disjoint hyperbolic communities, the clustering coefficients behave analogously. $T(i,j,h)$ and $W(i,j,h)$ need to be evaluated with respect to the community of the nodes. They must yield 0 in case i, j, and h belong to different communities and use the community specific parameters p_C and θ_C to evaluate 1_C otherwise. □

It is not trivial to analyse the effects of noise to the clustering coefficient. Triangles or wedges from the inside-community area disappear, and new ones get introduced involving the outside-community area. Given the overall density of a graph, the expected number of triangles or wedges is derivable, but integrating the specific intra-community structure into this expectation remains an open problem.

6 Empirical Parameter Distributions for HYGEN

In this section we detail how we obtained the suggested parameter distributions. We use four networks with community information from the Stanford Large Network Dataset collection [17]: Amazon, DBLP, Friendster, and YouTube. We draw a sample of 500 communities of size between 100 and 1000 nodes and compute the hyperbolic models for each community individually. This yields



(a) γ (b) H (c) size

Fig. 2: Fitting quality of the tested distribution functions compared to that of the chosen distribution for each observed parameter in each dataset.

empirical distributions of γ, H, and the (truncated) community size n_C. Note that the YouTube network has only 129 communities of the respective size.

For each of the empirical distributions for γ, H, and n_C, we fit a variety of distribution functions (Generalized extreme value, Inverse Gaussian, Birnbaum–Saunders, Exponential, log-Normal, log-Logistic, Gamma, Rayleigh, Weibull, Nakagami, Rician, Normal, Logistic, Extreme value, t-location-scale). Not every distribution is applicable for each of the parameters. While the observed γs look normally distributed, H and the community size show an exponential behaviour. Nevertheless, we evaluate the fitting quality in terms of negative log-likelihood (LL) for each possible distribution.

As we optimize the parameters of the distribution functions to explain the empirical observations, we notice that many of the compared distribution functions yield a similar shape. We compare the fit quality of the distribution we choose to that of every other possible distribution function. To do so, we subtract the negative LL of the chosen distribution from the negative LL of the remaining distributions. The results are summarized as boxplots in Fig. 2 for each data set. A value of 0 indicates similar quality. Higher values indicate that the chosen distribution function is a better fit, and lower values that another distribution fits the observations better.

For the empirical distributions of γ, we decide to propose the Normal distribution as a good description across the studied datasets. In terms of fit quality compared to the other distribution functions, this seems to be a favourable choice for the four datasets. The empirical distributions of the parameter H show more variation in their shape, their clear commonality is that thin tails are much more frequent than thick ones. Deriving the exact shape of the parameter distributions given these samples appears challenging. As Fig. 2b indicates, the distribution functions fit the data with varying quality. We choose to represent H by an Exponential function, but dependent on the data set there might be better options. For the community size, the Generalized extreme value distribution (GEV) yields the best fit in terms of LL for every dataset (see Fig. 2c).

7 HyGen Graphs from Known Distributions

While graphs modelled after empirical observations are potentially more realistic, HyGen can also be used to construct random graphs from pre-defined parameter distributions.

Suppose we aim to generate random graphs where the communities show a power law connectivity. This task might not seem not easily expressible in terms of γ and H, in particular because they are modelled independently. Recall that the hyperbolic model can be expressed with several equivalent formulations (see Section 3). If we express the model in terms of $\texttt{mixture}(x,\Sigma)$ an immediate solution is obvious: x becomes fixed to constantly 0 and only the distribution of the boundary condition Σ needs to be supplied, for instance in form of a Normal distribution.

Generating clique-like connectivity within the communities is as easy as setting γ and H constantly to their maximum, i.e. 100 % of the size of the community. Notice that the same result could however be achieved with less modelling effort.

8 Stability of the Graph Generation

With our analysis on real-world data, we demonstrate two contrasting aspects: here, we show how well HYGEN-generated graphs adapt to the characteristic distributions of these networks; in Section 9, we show that the graph generation procedure introduces enough randomness such that the resulting graph differs from its template.

To test HYGEN on real-world data sets[4], we fit distributions to the observed parameter distributions from every data set described in Section 6. With the obtained distribution functions we use HYGEN to generate new random graphs. On these new graphs, we compute the best hyperbolic model of each community. For comparison, we also evaluate the performance of LFR and DC-SBM on the task of generating graphs with hyperbolic communities. While LFR-generated graphs, like graphs from HYGEN, are equipped with ground-truth community information, the DC-SBM implementation[5] only derives the information about communities in the model fitting step. Hence, for this comparison we assume that DC-SBM correctly recovers communities in the graphs it has generated.

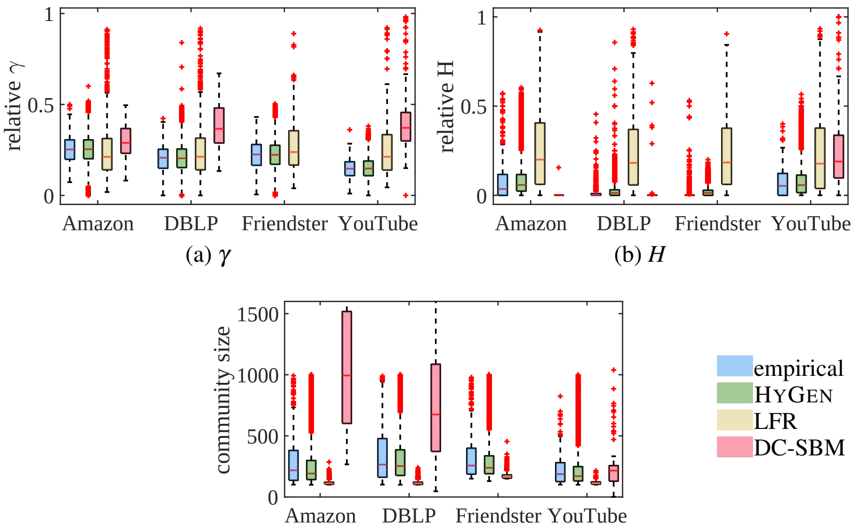
Fig. 3 shows boxplots of the empirical distributions of γ, H, and the community size in comparison to the distributions obtained after computing hyperbolic models on the results of HYGEN, LFR, and DC-SBM. Notice that 100 times as many random communities where drawn than what was in the input data, i.e. 12 900 communities for YouTube and 50 000 for the other datasets. For performance reasons, DC-SBM generated only one graph with 100 communities for every data set.

We observe a good resemblance of the median γs between the original and the HYGEN-generated graphs. For H, we observe mostly similar-looking distributions for the original communities and the HYGEN-generated ones. HYGEN however shows a tendency to produce communities with exceptionally thick tails. For the community size, Fig. 3c indicates, that although the medians of the generated graphs match the original, we have slightly less of the larger communities in the HYGEN-generated graphs. This is likely a result of the way we interpret the empirically observed distributions.

In the median, the shape of the communities LFR generates is quite accurate for the core size γ, however more communities with exceptionally large cores can be

[4] Our code: http://cs.uef.fi/~pauli/hybobo/rgg

[5] https://github.com/ntamas/blockmodel



(a) γ

(b) H

(c) community size (data for DC-SBM $>$ 1600 not displayed)

Fig. 3: Distributions of parameters in generated graphs compared to original data. H and γ are obtained after fitting hyperbolic models.

observed than in the original data.The height H of the tails is often overestimated. Community sizes are much smaller than in the empirical observation.

DC-SBM produces communities with thicker cores than originally observed, while the tail heights are underestimated, except in YouTube. Conversely, apart from YouTube, communities get much larger than originally observed, indicating that our assumption that DC-SBM correctly recovers communities in the graphs it generated may not hold. (Results for Friendster did not compute within a week.)

9 Randomness of the Generated Graphs

From the comparison of the parameter distributions of the original graphs and the generated graphs, we can conclude that their hyperbolic structure is similar. This however would also hold for an identical copy of the original graph. To assess whether the structurally similar graphs are also reasonably different to the original, we study their conditional entropy.

The conditional entropy $\mathscr{H}(Y|X)$ quantifies the amount of information needed to describe the outcome of a discrete random variable Y provided the discrete random variable X is known. The result is $\mathscr{H}(Y|X) = 0$ if Y is completely determined by X, and $\mathscr{H}(Y|X) = \mathscr{H}(Y)$ if and only if Y and X are independent [7]. As we are interested in the relative amount of dependency of Y given X, we scale $\mathscr{H}(Y|X)$ by $\mathscr{H}(Y)$ to obtain a value within $[0,1]$. We define the relative conditional entropy as $\mathscr{H}_{\mathrm{rel}}(Y|X) = \mathscr{H}(Y|X)/\mathscr{H}(Y)$. Since the graphs to compare are binary, $\mathscr{X} = \mathscr{Y} = \{0,1\}$. To interpret them as random variables X and Y, we vectorize the upper triangular of the adjacency matrices after ordering the nodes by their degree.

To compute $\mathscr{H}_{\mathrm{rel}}(Y|X)$, we generate 100 random graphs from the observed parameters of each dataset keeping the distribution of community sizes identical to the original graph to ensure that the resulting sample is of the same size. We compute $\mathscr{H}_{\mathrm{rel}}(Y|X)$ per community rather than for each entire generated graph for two reasons: First, the sizes of the communities are fixed for this experiment, which introduces an artificial amount of determinism that could bias the measurement. Second, from the original graphs, we sampled communities but their context is not maintained (in particular because we assume non-overlapping communities). Thus we would compare the inter-community structure of the original graph to uniformly at random distributed edges in the generated graphs. As our modelling focuses entirely on the intra-community structure, this comparison seems futile.
We obtain the following average relative conditional entropies over all the communities and all the samples: Amazon 0.996, DBLP 0.996, Friendster 0.990, YouTube 0.963 (standard deviations are within the displayed precision). This result indicates that the amount of determinism of the generated communities given the original is very small. Thus, we conclude that HYGEN generates structurally similar random graphs that still exhibit non-determinism compared to their template.

10 Conclusions

We have introduced the random graph generator HYGEN. Our results indicate that this generator is able to produce realistic intra-community structure, unlike the state-of-the-art methods. Despite simplifying assumptions, such as non-overlapping communities and a uniform noise model, HYGEN is a step towards more realistic random graph generators.
To further improve the modelling, important future work encompasses to handle overlapping communities as well as to offer more realistic noise models. While our current noise model assumes the average of the observed noise evenly for all communities, the examined data sets show evidence that noise is correlated with community size. Hence modelling noise in a more adjusted way could result in even more accurate models. In addition, while we estimate core sizes of communities precisely, covering in particular thin tails of them with higher accuracy could yield further improvement.
HYGEN has its obvious use for testing community detection algorithms. It can generate realistic graphs equipped with reliable labelling of the communities. Besides this, HYGEN might serve as an anonymization tool to study the structure of social networks without revealing the participants identities.

References

1. Aggarwal, C.C., Wang, H. (eds.): Managing and Mining Graph Data. Springer, New York (2010)
2. Alba, R.D., Moore, G.: Elite social circles. Sociol. Methods Res. **7**(2), 167–188 (1978)
3. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys **74**, 47–97 (2002)
4. Araujo, M., Günnemann, S., Mateos, G., Faloutsos, C.: Beyond blocks: Hyperbolic community detection. In: ECMLPKDD ’14, pp. 50–65 (2014)

5. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. Soc. Networks **21**, 375–395 (1999)
6. Chakrabarti, D., Zhan, Y., Faloutsos, C.: R-MAT: A Recursive Model for Graph Mining. In: SDM '04, pp. 442–446 (2004)
7. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, Hoboken, NJ (2006)
8. Erdős, P., Rényi, A.: On random graphs I. Publi. Math. Debrecen **6**, 290 (1959)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Nat. Acad. Sci. **99**(12), 7821–7826 (2002)
10. Holland, P., Laskey, K., Leinhardt, S.: Stochastic blockmodels: First steps. Soc. Networks **5**(2), 109–137 (1983)
11. Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. Phys. Rev. E **83**, 016,107 (2011)
12. Knuth, D.E.: The Art of Computer Programming Vol. 2: Seminumerical Algorithms, 2 edn. Addison-Wesley, Reading, MA (1981)
13. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., Boguñá, M.: Hyperbolic geometry of complex networks. Phys. Rev. E **82**, 036,106 (2010)
14. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046,110 (2008)
15. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: An approach to modeling networks. J. Mach. Learn. Res. **11**, 985–1042 (2010)
16. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: Densification laws, shrinking diameters and possible explanations. In: KDD '05, pp. 177–187 (2005)
17. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (2014). Accessed 11 Feb 2016.
18. Metzler, S., Günnemann, S., Miettinen, P.: Hyperbolae are no hyperbole: Modelling communities that are not cliques. In: ICDM '16, pp. 330–339 (2016)
19. Morgan, D.L., Neal, M.B., Carder, P.: The stability of core and peripheral networks over time. Soc. Networks **19**(1), 9–25 (1997)
20. Orbanz, P., Roy, D.M.: Bayesian models of graphs, arrays and other exchangeable random structures. IEEE Trans. Patern. Anal. **37**(2), 437–461 (2015)
21. Panzarasa, P., Opsahl, T., Carley, K.M.: Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. J. Am. Soc. Inf. Sci. Technol. **60**(5), 911–932 (2009)
22. Park, H., Kim, M.S.: EvoGraph: An Effective and Efficient Graph Upscaling Method for Preserving Graph Properties. In: KDD '18, pp. 2051–2059 (2018)
23. Reed, P.B., Selbee, L.K.: The Civil Core in Disproportionality in Charitable Giving, Volunteering, Civic Participation. Nonprofit Volunt. Sect. Q. **30**(4), 761–780 (2001)
24. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
25. Zhang, J.W., Tay, Y.C.: GSCALER: Synthetically Scaling A Given Graph. In: EDBT'16, pp. 53–64 (2016)
26. Zhu, Y., Yan, X., Moore, C.: Oriented and degree-generated block models: generating and inferring communities with inhomogeneous degree distributions. J. Complex Networks **2**(1), 1–18 (2014)