Outliers in the ABCD Random Graph Model with Community Structure (ABCD+o)

Bogumił Kamiński[1], Paweł Prałat[2], and François Théberge[3]

[1] SGH Warsaw School of Economics, Warsaw, Poland,
e-mail: bkamins@sgh.waw.pl,
[2] Toronto Metropolitan University, Toronto, ON, Canada,
e-mail: pralat@ryerson.ca,
[3] The Tutte Institute for Mathematics and Computing, Ottawa, ON, Canada,
e-mail: theberge@ieee.org

Abstract. The **A**rtificial **B**enchmark for **C**ommunity **D**etection graph (**ABCD**) is a random graph model with community structure and power-law distribution for both degrees and community sizes. The model generates graphs with similar properties as the well-known **LFR** one, and its main parameter ξ can be tuned to mimic its counterpart in the **LFR** model, the mixing parameter μ.
In this paper, we extend the **ABCD** model to include potential outliers. We perform some exploratory experiments on both the new **ABCD+o** model as well as a real-world network to show that outliers posses some desired, distinguishable properties.

Keywords: ABCD model, outliers, community detection

1 Introduction

One of the most important features of real-world networks is their community structure, as it reveals the internal organization of nodes [7]. In social networks communities may represent groups by interest, in citation networks they correspond to related papers, in the Web communities are formed by pages on related topics, etc. Being able to identify communities in a network could help us to exploit this network more effectively. Grouping like-minded users or similar-looking items together is important for a wide range of applications including recommendation systems, anomaly or outlier detection, fraud detection, rumour or fake news detection, etc. [10]. For more discussion around various aspects of mining complex networks see, for example, [19, 14].

It was identified as one of the major current challenges in detecting communities that most of the existing algorithms treat all nodes the same way, that is, they try to assign them to precisely one community. On the other hand, many complex networks (regardless whether their nodes correspond to, say, users of some social media or movies on Netflix) consist of nodes that are more active participants of their own communities while others are not [17]. As a result, there is a need to detect outlier nodes that are not part of any of the communities.

Moreover, some communities might be overlapping which is reflected by some of the nodes belonging to a few communities via fuzzy membership. Some recent algorithms (see, for example [8, 2] or **NI-Louvain** [22]) try to incorporate these notions but more research is expected to be pursued in the near future. For more on anomalies and outliers in graphs see, for example, the survey [1].

Another well-known challenge recognized by many researchers is that there are very few datasets with ground-truth identified and labelled. As a result, there is need for synthetic random graph models with community structure that resemble real-world networks in order to benchmark and tune clustering algorithms that are unsupervised by nature. The **LFR** (Lancichinetti, Fortunato, Radicchi) model [16, 15] generates networks with communities and at the same time it allows for the heterogeneity in the distributions of both node degrees and of community sizes. It became a standard and extensively used method for generating artificial networks with (non-overlapping) community structure.

Unfortunately, the situation is much more challenging if one needs a synthetic model with outliers. There seems to be no standard model that one may use. For example, in [8] the authors adjust the classical Stochastic Block Model to simultaneously take into account the community structure and outliers by introducing different probabilities of connection between inliers and pairs involving outliers. To validate algorithms tested in [2], the authors start with a synthetic **LFR** network or a real-world one and then randomly perturb edges around some randomly selected nodes in order to create artificial outliers. **LFR** itself [15] has some basic functionality to create overlapping clusters but not outliers.

In this paper, we revisit the Artificial Benchmark for Community Detection (**ABCD** graph) [13] that was recently introduced and implemented[4], including a fast implementation that uses multiple threads (**ABCDe**) [11][5]. Undirected variant of **LFR** and **ABCD** produce graphs with comparable properties but **ABCD**/**ABCDe** is faster than **LFR** and can be easily tuned to allow the user to make a smooth transition between the two extremes: pure (disjoint) communities and random graph with no community structure. Moreover, it is easier to analyze theoretically. For example, various theoretical asymptotic properties of the **ABCD** model are analyzed in [12], including the modularity function that is, arguably, the most important graph property of networks in the context of community detection.

We extend the original **ABCD** model to include potential outliers (see Section 2). We examine one of the few real-world networks with identified outliers, the College Football Graph (see Subsection 3.1), and identify a few distinctive properties of outliers that are present in this network. We then perform a few simulations with our new **ABCD+o** model to show that its outliers posses similar properties (see Subsections 3.2 and 3.3). Future directions are briefly mentioned in Section 4.

[4] https://github.com/bkamins/ABCDGraphGenerator.jl/
[5] https://github.com/tolcz/ABCDeGraphGenerator.jl/

2 Adjusting the ABCD Model to Include Outliers

We start this section with a brief description of the **ABCD** model taken from [11]; details can be found in [13] or in [12]. We then carefully explain the adjustments needed to incorporate the existence of outliers.

2.1 The Original Model

As in **LFR** model [16, 15], for a given number of nodes n, we start by generating a power law distribution both for the degrees and community sizes. Those are governed by the power law exponent parameters (γ, β). We also provide additional information to the model, again as it is done in **LFR**, namely, the average and the maximum degree, and the range for the community sizes. The user may alternatively provide a specific degree distribution and/or community sizes.

For each community, we generate a random *community* subgraph on the nodes from a given community using either the **configuration model** [4] (see [3, 23, 24] for related models and results) which preserves the exact degree distribution, or the **Chung-Lu model** [5] which preserves the expected degree distribution. On top of it, we independently generate a *background* random graph on all the nodes. Everything is tuned properly so that the degree distribution of the union of all graphs follows the desired degree distribution (only in expectation in the case of the Chung-Lu variant). The mixing parameter ξ guides the proportion of edges which are generated via the background graph. In particular, in the two extreme cases, when $\xi = 1$ the graph has no community structure while if $\xi = 0$, then we get disjoint communities. In order to generate simple graphs, we may have to do some re-sampling or edge re-wiring, which are described in [13].

During this process, larger communities will additionally get some more internal edges due to the background graph. As argued in [13], this "global" variant of the model is more natural and so we recommend it. However, in order to provide a variant where the expected proportion of internal edges is exactly the same for every community (as it is done in **LFR**), we also provide a "local" variant of **ABCD** in which the mixing parameter ξ is automatically adjusted for every community.

Two examples of **ABCD** graphs on $n = 100$ nodes are presented in Figure 1. Degree distribution was generated with power law exponent $\gamma = 2.5$ with minimum and maximum values 5 and 15, respectively. Community sizes were generated with power law exponent $\beta = 1.5$ with minimum and maximum values 20 and 40, respectively; communities are shown in different colours. The global variant and the configuration model was used to generate the graphs. The left plot has the mixing parameter set $\xi = 0.2$ while the "noisier" graph on the right plot has the parameter fixed to $\xi = 0.4$.

2.2 Adjusting the Model to Include Outliers

The adjusted model, **ABCD+o** (**ABCD** with **outliers**), will have additional parameter s_0 which is equal to the number of outliers. Because of a well structured and flexible design of the original model, adjusting it to include outliers

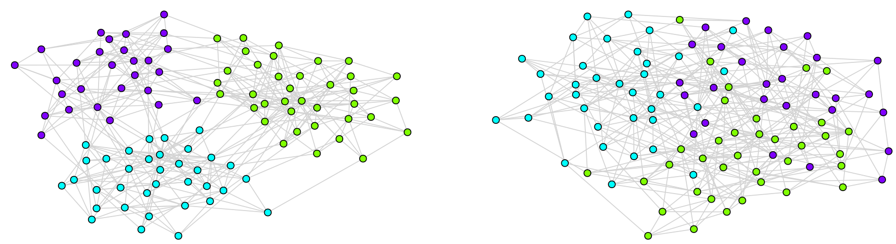


Fig. 1: Two examples of **ABCD** graphs with low level of noise ($\xi = 0.2$, left) and high level of noise ($\xi = 0.4$, right).

is simple. One trivial adjustment needed is in the way the distribution of community sizes is generated. Slightly more delicate modification is needed in the process of assigning nodes to communities. However, before that the algorithm needs to select suitable nodes for outliers. Below, we independently discuss these issues and explain how they are generalized.

The **ABCD+o** extension is defined only for the default settings of the original **ABCD** algorithm, namely, for the global version of the algorithm, configuration model used to generate community and background graphs, and accepts only parameter ξ as the level of noise.

Distribution of Community Sizes. As in the original **ABCD** model, the degree distribution is generated randomly following the (truncated) *power-law distribution* $\mathcal{P}(\gamma, \delta, \Delta)$ with exponent γ, minimum value δ, and maximum value Δ. Let $\beta \in \mathbf{R}_+$, $s, S \in \mathbf{N}$ such that $\delta < s \le S$. It is recommended to use $\beta \in (1, 2)$, some relatively small value of s such as 100 or 500, and S larger than Δ. The condition for S is needed to make sure large degree nodes have large enough communities to be assigned to. Similarly, the assumption that $s \ge \delta + 1$ is required to guarantee that small communities are not too small and so that they can accommodate small degree nodes. These conditions are needed to make sure that generating a simple graph with the desired properties is feasible.

Community sizes in the original **ABCD** model are generated randomly following the (truncated) *power-law distribution* $\mathcal{P}(\beta, s, S)$ with exponent β, minimum value s, and maximum value S. Communities are generated with this distribution as long as the sum of their sizes is less than n, the desired number of nodes. After drawing a predetermined number of samples from this distribution, the algorithm is selecting one sequence with the sum as close to n as possible and carefully adjusts it, if needed.

Since there are s_0 outliers in the new model, the community sizes (s_i, $i \in [\ell] := \{1, \dots, \ell\}$) are generated as in the original model but this time with the condition that the sum of their sizes is equal to $n - s_0$ (instead of n).

Assigning Nodes to Outliers. Parameter $\xi \in (0,1)$ reflects the amount of noise in the network. It controls the fraction of edges that are between communities. Indeed, in the original **ABCD** model, asymptotically (but not exactly) $1-\xi$ fraction of edges end up within one of the communities. Each node in the original model has its degree w_i split into two parts: *community degree* y_i and *background degree* z_i ($w_i = y_i + z_i$). The goal is to get $y_i \approx (1-\xi)w_i$ and $z_i \approx \xi w_i$. However, both y_i and z_i have to be non-negative integers and for each community $C \subseteq V$, $\sum_{i \in C} y_i$ has to be even. Fortunately, this can be easily achieved by an appropriate random rounding of $(1-\xi)w_i$ to the nearest integers.

In the generalized **ABCD+o** model, each non-outlier has its degree w_i split into y_i and z_i, as in the original model. These nodes will be assigned into one community. On the other hand, outliers will not get assigned to any community and all of their neighbours will be in the background graph and so they will be "sprinkled" across the whole graph. As a result, their degrees will satisfy $w_i = z_i$. Note that the only potential problem with outliers that might occur is when ξ is close to zero. At the extreme case when $\xi = 0$, only outliers have non-zero degree in the background graph. In order to make sure that there exists a simple graph that satisfies the required degree distribution, in such extreme situations all outliers must have degrees smaller than s_0. The model needs to be prepared for such potential problems but in practice (when the number of nodes n is large, the number of outliers s_0 is relatively small, and the level of noise ξ is not zero) there are plenty of nodes with non-zero degree in the background graph and so there is no restriction for outliers.

To prepare for a potential problem we do the following. Once the degree of each node w_i is split into y_i and z_i, we get a lower bound for the number of nodes that will have non-zero degree in the background graph, namely, $L := |\{v \in V : z_i \geq 1\}|$. Note that $\bar{L} = \mathbf{E}[L] = \sum_{i \in V} \min(1, \xi w_i)$ since each node with $\xi w_i \geq 1$ satisfies $z_i \geq 1$ and each node with $\xi w_i < 1$ has $z_i = 1$ with probability ξw_i and $z_i = 0$ otherwise. Moreover, since by default outliers have $z_i = w_i \geq 1$, there will be at least s_0 vertices of positive degree in the background graph. Assuming that outliers are selected uniformly at random, we expect $L + (n-L)(s_0/n)$ nodes of positive degree in the background graph. (In fact, since there is a slight bias toward selecting small degree nodes for outliers and L has a bias toward large degree nodes, we expect slightly more nodes of positive degree in the background graph, which is good.) We introduce the following constraint: a node of degree w_i can become an outlier if

$$w_i \leq \bar{L} + s_0 - \bar{L}s_0/n - 1. \tag{1}$$

Finally, s_0 nodes satisfying (1) are selected uniformly at random to become outliers. (In the implementation, these nodes simply form an independent "community" with $y_i = 0$ and $z_i = w_i$.)

Assigning Nodes to Communities. Similarly to the potential problem with outliers, we need to make sure that non-outliers of large degree are not assigned to small communities. Based on the parameter ξ we know that roughly $(1-\xi)w_i$

neighbours of a node of degree w_i will be present in its own community. However, this is only the lower bound as some neighbours in the background graph might end up there by chance. Hence, in order to make enough room in the community graph for all neighbours of a given node, the original **ABCD** algorithm needs to compute x_i, the expected number of neighbours of a node of degree w_i that end up in its own community. We need to recompute x_i to incorporate the existence of outliers.

Assuming that nodes are assigned randomly with a distribution close to the uniform distribution, we expect Ws_0/n points (in the corresponding configuration model) in the background graph to be associated with outliers, where $W := \sum_{i \in [n]} w_i$ is the volume of the graph (equivalently, the total number of points in the corresponding configuration model). Similarly, we expect ξ fraction of the points associated with non-outliers to end up in the background graph, that is, $W(1-s_0/n)\xi$ points. In order to estimate what fraction of neighbours of a given non-outlier node is expected to be within the same community, we need to answer the following question: what is the probability that a random point in the background graph associated with a non-outlier is matched with a point within the same community? It is equal to

$$\sum_{j \in [\ell]} \frac{s_j}{n-s_0} \cdot \frac{\frac{s_j}{n-s_0} W(1-s_0/n)\xi}{W(1-s_0/n)\xi + Ws_0/n} = \sum_{j \in [\ell]} \left(\frac{s_j}{n-s_0} \right)^2 \frac{(n-s_0)\xi}{(n-s_0)\xi + s_0}.$$

Indeed, with probability $\frac{s_j}{n-s_0}$ a random point belongs to community j. There are $\frac{s_j}{n-s_0} W(1-s_0/n)\xi$ points associated with community j and the total number of points in the background graph is $W(1-s_0/n)\xi + Ws_0/n$. Hence, one can easily estimate the probability that the point from community j is matched with another point from the same community. The expected number of neighbours of a node of degree w_i that stay within the same community is then

$$x_i := \left(1 - \xi + \xi \sum_{j \in [\ell]} \left(\frac{s_j}{n-s_0} \right)^2 \frac{(n-s_0)\xi}{(n-s_0)\xi + s_0} \right) w_i = (1-\xi\phi) w_i,$$

where

$$\phi := 1 - \sum_{j \in [\ell]} \left(\frac{s_j}{n-s_0} \right)^2 \frac{(n-s_0)\xi}{(n-s_0)\xi + s_0}.$$

In particular, we expect $(1-\xi\phi)(1-s_0/n)$ fraction of edges to stay within one of the communities. Moreover, as expected, if $s_0 = 0$, then we recover the value of ϕ used in the original **ABCD** model, namely,

$$\phi = 1 - \sum_{j \in [\ell]} \left(\frac{s_j}{n} \right)^2.$$

As in the original **ABCD** model, a node of degree w_i can be assigned to community of size s_j if $x_i \le s_j - 1$. We select one admissible assignment of non-outliers to communities uniformly at random which turns out to be relatively easy from both theoretical and practical points of view.

Two examples of **ABCD+o** graphs on $n = 100$ nodes are presented in Figure 2. The number of outliers is $s_0 = 5$ and the remaining parameters are exactly the same as the ones to produce Figure 1. Communities are shown in different colours and outliers are displayed with triangular shape. The left plot has the mixing parameter set $\xi = 0.2$ while the "noisier" graph on the right plot has the parameter fixed to $\xi = 0.4$. In the left plot it is visible that 4 out of 5 outliers are clearly located *between* the communities (one of them is within a community as outlier can, by pure chance, get many edges within one community). In the right plot, which is more noisy, we still see that outliers are surrounded by nodes belonging to different communities.

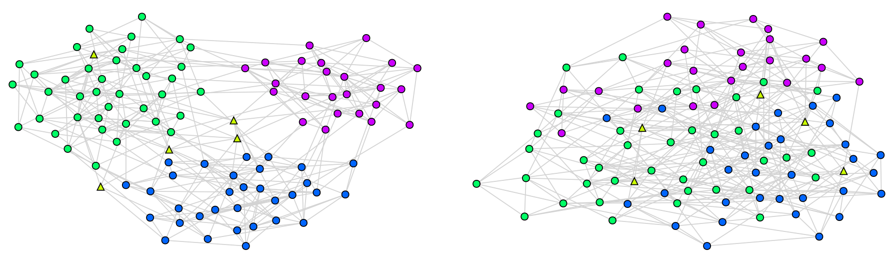


Fig. 2: Two examples of **ABCD+o** graphs with low level of noise ($\xi = 0.2$, left) and high level of noise ($\xi = 0.4$, right). The number of outliers is $s_0 = 5$.

3 Experiments—Distinguishing Properties of Outliers

In order to better understand properties of outliers, we perform a few simple and exploratory experiments on the well-known College Football real-world network with known community structure and the presence of outliers. We identified three natural properties that distinguish outliers from non-outliers.

In order to show that our new **ABCD+o** model exhibits similar desired properties, we generated graphs on $n = 10{,}000$ nodes and $s_0 = 500$ outliers (5%). Degree distribution was generated with power law exponent $\gamma = 2.5$ with minimum and maximum values 5 and 500, respectively. Community sizes were generated with power law exponent $\beta = 1.5$ with minimum and maximum values 100 and 1,000, respectively. We independently generated graphs for all values of $\xi \in \{0.0, 0.1, \ldots, 1.0\}$ but the degree distribution and the distribution of community sizes were coupled (it is easy to do in our implementation) so that all 11 graphs use the same distributions.

3.1 The College Football Graph

The College Football real-world network represents the schedule of United States football games between Division IA colleges during the regular season in Fall 2000 [9]. The data consists of 115 teams (nodes) and 613 games (edges). The teams are divided into conferences containing 8–12 teams each. In general, games are more frequent between members of the same conference than between members of different conferences, with teams playing an average of about seven intra-conference games and four inter-conference games in the 2000 season. There are a few exceptions to this rule, as detailed in [18]: one of the conferences is really a group of independent teams, one conference is really broken into two groups, and 3 other teams play mainly against teams from other conferences. We refer to those 14 teams as outlying nodes, which we represent with a distinctive triangular shape in Figure 3.

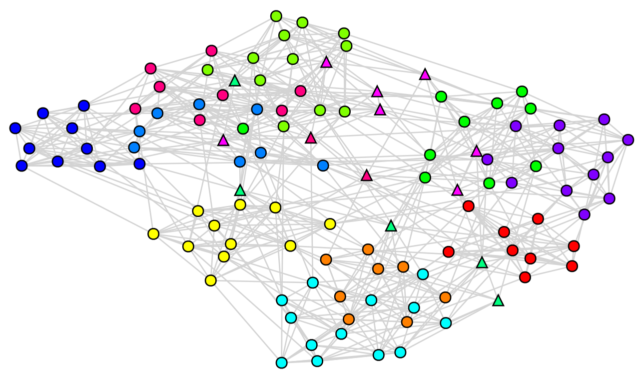


Fig. 3: The College Football Graph; outliers are displayed with triangular shape.

3.2 Participation Coefficient

The following definitions are commonly used in the literature [6, 21] (see also [14]). We say that a set of nodes $C \subseteq V$ forms a *strong community* if each node in C has more neighbours in C than outside of C. One may relax this strong notion and say that C forms a *weak community* if the average degree inside the community C (over all nodes in C) is larger than the corresponding average number of neighbours outside of C. In this context, an *outlier* could be formally defined as a node that does not have majority of its neighbours in any of the communities. In the **ABCD+o** model, non-outliers are expected to have more than half of their neighbours in its own community, provided that $\xi < 0.5$. On the other hand, outliers are expected to satisfy the desired property, unless there is an enormous community spanning more than 50% of nodes.

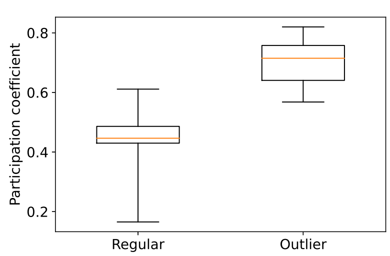
A more refined picture is provided by the next coefficient that is a natural measure of concentration. For any partition $\mathbf{A} = \{A_1, \ldots, A_\ell\}$ of the set of nodes, the *participation coefficient* of a node v (with respect to $\mathbf{A}$) is defined as follows:

$$p(v) = 1 - \sum_{i=1}^{\ell} \left(\frac{\deg_{A_i}(v)}{\deg(v)} \right)^2,$$

where $\deg_{A_i}(v)$ is the number of neighbours of v in A_i. The participation coefficient $p(v)$ is equal to zero if v has neighbours exclusively in one part. Members of strong communities satisfy, by definition, $p(v) < 3/4$. In the other extreme case, the neighbours of v are homogeneously distributed among all parts and so $p(v)$ is close to the trivial upper bound of

$$1 - \sum_{i=1}^{\ell} \left(\frac{\deg(v)/\ell}{\deg(v)} \right)^2 = 1 - \frac{1}{\ell} \approx 1.$$

For the experiments shown below, even though we have the ground truth communities available to use, we computed the participation coefficients using communities (partition $\mathbf{A}$) we obtained with the **ECG** clustering algorithm which we describe in the following subsection. The distribution of the participation coefficient among outliers and non-outliers for the College Football Graph is presented on box plot in Figure 4 (left). We see that outliers have significantly larger average value of $p(v)$ than the corresponding value for non-outliers: 0.709 vs. 0.439. The corresponding averages (together with associated standard deviations) for the **ABCD+o** model with different level of noise are presented in Figure 4 (right). For low level of noise (small values of ξ) there is a clear difference between outliers and non-outliers but the discrepancy diminishes for noisy graphs (large values of ξ). In the extreme case when $\xi = 1$ there is no difference between the two classes and so the averages are close to each other as they should.



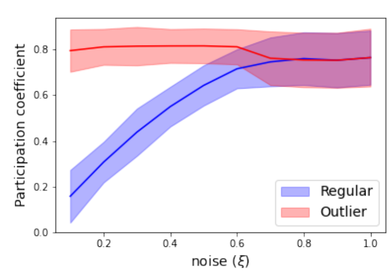


Fig. 4: Distribution of the participation coefficient for regular and outlier nodes: College Football Graph (left) and **ABCD+o** model (right).

3.3 ECG Votes

Ensemble Clustering algorithm for Graphs (**ECG**) [20][6] is a consensus clustering method based on the classical **Louvain** algorithm. In its first phase, several low-level partitions are computed with different randomization, and for each edge the proportion of times both nodes ended up in the same part is computed. Those are the *ECG edge scores*. High scores are indicative of stable pairs that often appear in the same part. For a given node v, we define $E(v)$ to be the average ECG score over all edges incident to v, and we call it the *ECG coefficient* of a node v. It is expected that outliers are more challenging to cluster which should be manifested by relatively small ECG coefficients $E(v)$ associated with these nodes.

As it was done for the participation coefficient, we investigate the distribution of the ECG coefficient among outliers and non-outliers for the College Football Graph—see Figure 5 (left). We see that it is another distinguishing coefficient—outliers have significantly smaller average value of $E(v)$ than the corresponding value for non-outliers: 0.465 vs. 0.701. Similar conclusions can be derived from the corresponding averages for the **ABCD+o** model—see Figure 5 (right). As before, the difference becomes less visible as more noise is introduced.

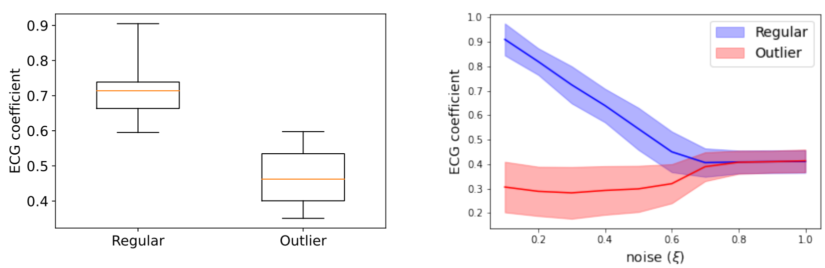


Fig. 5: Distribution of the ECG coefficient for regular and outlier nodes: College Football Graph (left) and **ABCD+o** model (right).

4 Future Directions

In this paper, we extended the **ABCD** model to **ABCD+o** which incorporates the presence of outliers. We investigated a few properties that are able to distinguish outliers from regular nodes. One may try to extend these ideas further and build an outlier detection algorithm. Another important extension of the original **ABCD** that we leave for the future is to design a variant of the model to include overlapping clusters. An orthogonal future direction that we

[6] https://github.com/ftheberge/graph-partition-and-measures

(and industry partners that we collaborate with) are interested in is to design a hypergraph model with known community structure.

References

1. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data mining and knowledge discovery 29(3), 626–688 (2015)
2. Bandyopadhyay, S., Vivek, S.V., Murty, M.N.: Integrating network embedding and community outlier detection via multiclass graph description. arXiv preprint arXiv:2007.10231 (2020)
3. Bender, E.A., Canfield, E.R.: The asymptotic number of labeled graphs with given degree sequences. Journal of Combinatorial Theory, Series A 24(3), 296–307 (1978)
4. Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. European Journal of Combinatorics 1(4), 311–316 (1980)
5. Chung Graham, F., Lu, L.: Complex graphs and networks. No. 107, American Mathematical Soc. (2006)
6. Flake, G.W., Lawrence, S., Giles, C.L.: Efficient identification of web communities. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 150–160 (2000)
7. Fortunato, S.: Community detection in graphs. Physics reports 486(3-5), 75–174 (2010)
8. Gaucher, S., Klopp, O., Robin, G.: Outlier detection in networks with missing links. Computational Statistics & Data Analysis 164, 107308 (2021)
9. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the national academy of sciences 99(12), 7821–7826 (2002)
10. Javed, M.A., Younis, M.S., Latif, S., Qadir, J., Baig, A.: Community detection in networks: A multidisciplinary review. Journal of Network and Computer Applications 108, 87–111 (2018)
11. Kamiński, B., Olczak, T., Pankratz, B., Prałat, P., Théberge, F.: Properties and performance of the abcde random graph model with community structure. arXiv:2203.14899 (2022)
12. Kamiński, B., Pankratz, B., Prałat, P., Théberge, F.: Modularity of the abcd random graph model with community structure. arXiv:2203.01480 (2022)
13. Kamiński, B., Prałat, P., Théberge, F.: Artificial benchmark for community detection (abcd)—fast random graph model with community structure. Network Science pp. 1–26 (2021)
14. Kamiński, B., Prałat, P., Théberge, F.: Mining complex networks (2021)
15. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80(1), 016118 (2009)
16. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Physical review E 78(4), 046110 (2008)
17. Liu, F., Wang, Z., Deng, Y.: Gmm: A generalized mechanics model for identifying the importance of nodes in complex networks. Knowledge-Based Systems 193, 105464 (2020)
18. Lu, Z., Wahlström, J., Nehorai, A.: Community detection in complex networks via clique conductance. Scientific reports 8(1), 1–16 (2018)
19. Newman, M.E.J.: Networks (2nd edition). Oxford University Press, Oxford; New York (2018)

20. Poulin, V., Théberge, F.: Ensemble clustering for graphs. In: International Conference on Complex Networks and their Applications. pp. 231–243. Springer (2018)
21. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the national academy of sciences 101(9), 2658–2663 (2004)
22. Singh, D., Garg, R.: Ni-louvain: A novel algorithm to detect overlapping communities with influence analysis. Journal of King Saud University-Computer and Information Sciences (2021)
23. Wormald, N.C.: Generating random regular graphs. Journal of algorithms 5(2), 247–280 (1984)
24. Wormald, N.C., et al.: Models of random regular graphs. London Mathematical Society Lecture Note Series pp. 239–298 (1999)