Improving community detection via community association strength scores

Jordan Barrett[1], Ryan DeWolfe[1], Bogumił Kamiński[2], Paweł Prałat[1], Aaron Smith[3], and François Théberge[4]

[1] Dept of Mathematics, Toronto Metropolitan University, Toronto, Canada; {jordan.barrett, ryan.dewolfe, pralat}@torontomu.ca

[2] Decision Analysis and Support Unit, SGH Warsaw School of Economics, Warsaw, Poland; bkamins@sgh.waw.pl

[3] Dept of Mathematics and Statistics, University of Ottawa, Ottawa, Canada; asmi28@uOttawa.ca

[4] Tutte Institute for Mathematics and Computing, Ottawa, Canada; theberge@ieee.org

Abstract. Community detection methods play a central role in understanding complex networks by revealing highly connected subsets of entities. However, most community detection algorithms generate partitions of the nodes, thus (i) forcing every node to be part of a community and (ii) ignoring the possibility that some nodes may be part of multiple communities. In our work, we investigate three simple community association strength (CAS) scores and their usefulness as post-processing tools given some partition of the nodes. We show that these measures can be used to improve node partitions, detect outlier nodes (not part of any community), and help find nodes with multiple community memberships.

1 Introduction

Detecting and analyzing the community structure in graphs is a fundamental problem in applied graph theory. Algorithms such as Girvan-Newman [22], Louvain [2], and Leiden [27] attempt to partition the nodes of a graph into clusters with the general goal of maximizing edge density within clusters and minimizing edge density between clusters. These global partitioning algorithms have been used to detect community structure in various real-world networks such as collaboration networks [22,20], biological networks [26,3], and social media networks [9,23], and have been tested and studied on synthetic models with ground-truth communities such as ABCD [11] and LFR [18]. Typically, these community detection algorithms return a partition, meaning each node must be part of exactly one community. In many real-world networks, though, members can belong to 0, 1, or multiple communities. Some algorithms to detect overlapping communities include clique-percolation [5], edge clustering [1] and ego-split [6]. A post-processing algorithm was proposed [10] to detect overlapping communities, but this approach is based on k-core clustering and often only covers a small part of the nodes.

In this paper, we use local community aware scores called *community association strength* (CAS) scores as a post processing tool for improving community detection in networks. These CAS scores are inspired by the measures proposed in [12]. The rest of the paper is organized as follows. In Section 2, we outline three CAS scores and show their similarities, differences, and relative abilities to predict community involvement. In Section 3, we show how such CAS scores can be used for three different but complementary tasks: (i) improving an existing graph partitioning algorithm, (ii) detecting nodes not part of any community (outlier nodes), and (iii) detect multi-community node memberships. Finally, some concluding remarks are given in Section 4

2 Community association strength

2.1 CAS scores

Given a graph $G = (V, E)$, a CAS score is any function $f : V \times 2^V \to [0, 1]$. In practice, we aim to find a CAS score f such that $f(v, C)$ indicates how well v is associated to community C, i.e., $f(v, C) \approx 1$ should imply that v is strongly associated to community C whereas $f(v, C) \approx 0$ should imply that v has little to no association to community C. We will motivate and outline three such functions.

Internal edge fraction (IEF) Arguably, the simplest measure for how strongly v is associated with community C is the *internal edge fraction.* For a community $C \subseteq V$, write $\deg_C(v) := |\{(u, v) \in E : u \in C\}|$ and $\deg(v) := \deg_V(v)$. Then, the internal edge fraction of v in C is defined as

$$\mathrm{IEF}(v, C) := \frac{\deg_C(v)}{\deg(v)} \,.$$

Although the internal edge fraction is both easy to interpret and a strong indicator of community association, it fails to account for the size of the communities. For example, if G contains a large community C_1 with, say, $|C_1| \approx |V|/2$ and a small community C_2 with $|C_2| \ll |V|$, then we may want to distinguish the outcomes $\mathrm{IEF}(v, C_1) = 1$ and $\mathrm{IEF}(v, C_2) = 1$, especially if $\deg(v)$ is large.

Normalized internal edge fraction (NIEF) For a graph $G = (V, E)$ and a community C, write $w(C) := \mathrm{vol}\,(C)/\mathrm{vol}\,(V)$ where $\mathrm{vol}\,(C) := \sum_{v \in C} \deg(v)$. The second CAS score we consider is the *normalized internal edge fraction*, defined as

$$\mathrm{NIEF}(v, C) := \max\left\{\mathrm{IEF}(v, C) - w(C), 0\right\} \,.$$

This CAS score is derived as follows. Let $G = (V, E)$, let $\mathbf{d} = \{\deg(v), v \in V\}$ and let $\widehat{G} \sim \mathrm{ChungLu}(\mathbf{d})$ [4]. Then, for any $v \in V$,

$$\mathrm{NIEF}(v, C) := \max\left\{\mathrm{IEF}_G(v, C) - \mathbb{E}\left[\mathrm{IEF}_{\widehat{G}}(v, C)\right], 0\right\} \,,$$

where we assume that C is fixed before sampling $\widehat{G}$. In other words, $\mathrm{NIEF}(v,C)$ compares the actual internal edge fraction of v to the "expected" such fraction under the associated null-model. Note that this score is inspired by the *community association strength* feature presented in [12].

Consider again the previous example with G containing communities C_1 and C_2 with $|C_1| \approx |V|/2$ and $|C_2| \ll |V|$. If $\mathrm{IEF}(v,C_1) = 1$ and $\mathrm{IEF}(v,C_2) = 1$ then we get that $\mathrm{NIEF}(v,C_1) \approx 0.5$ and $\mathrm{NIEF}(v,C_2) \approx 1$. These scores agree with the intuition that v having many connections into a small community is more surprising than having the same number of connections into a large community.

P score Finally, we consider a score based on the classic p-value significance test. Write $F(\cdot;n,p)$ for the CDF of the distribution $\mathrm{Binomial}(n,p)$. The P score is defined as

$$\mathrm{P}(v,C) := F\left(\deg_C(v) - 1; \deg(v), w(C)\right) .$$

Here, $1 - \mathrm{P}(v,C)$ is the probability that at least $\deg_C(v)$ edges join v and C in a Chung-Lu resampling of G (keeping C fixed), i.e., $1 - \mathrm{P}(v,C)$ is the classic p-value of $\deg_C(v)$ when considering the Chung-Lu model as the null model. Similar to the NIEF score, the P score prefers smaller communities. We will explore the similarities and differences between the three measures in the next section. Note that the P score is similar to, albeit distinct from, the community fitness measure used in [19].

These three measures are by no means a comprehensive list of CAS scores. Rather, we have chosen these scores to show how CAS can be used to analyse graphs with community structure containing outliers and overlapping communities.

2.2 Comparing CAS scores

All three CAS scores, IEF, NIEF, and P, share two qualitative properties that we believe are good indicators of the association strength of a node into a community. Firstly, all three scores evaluate 0 if no edges join v and community C. Secondly, for fixed $\mathrm{vol}\,(C)$, all three scores are monotone with respect to $\deg_C(v)$; if $\mathrm{vol}\,(C_1) = \mathrm{vol}\,(C_2)$ and $\deg_{C_1}(v) > \deg_{C_2}(v)$ then (v, C_1) yields a higher CAS score than (v, C_2) under all three measures.

Perhaps more interesting than the similarities between these three CAS scores are their differences. A key difference is how the measures change based on the community size and the degree of the node. We will highlight this difference by comparing the measures under two different transforms. Given a graph $G = (V, E)$ and a community C, consider the following transforms.

1. Add a disconnected k-clique to the graph and assign each new vertex to community C.
2. Create a second copy of each edge in E.

Now let G be a graph, C be a community in G, v be a vertex in C, and let G_1 and G_2 be constructed from G by the respective transforms. Firstly, we have that $\mathrm{IEF}_{G_1}(v, C) = \mathrm{IEF}_{G_2}(v, C) = \mathrm{IEF}_G(v, C)$, as the fraction of edges into C has not changed under either transform. Next, note that $\mathbb{E}\left[\mathrm{IEF}_{G_1}(v, C)\right] > \mathbb{E}\left[\mathrm{IEF}_G(v, C)\right]$ whereas $\mathbb{E}\left[\mathrm{IEF}_{G_2}(v, C)\right] = \mathbb{E}\left[\mathrm{IEF}_G(v, C)\right]$, implying that $\mathrm{NIEF}_{G_1}(v, C) < \mathrm{NIEF}_G(v, C)$ and $\mathrm{NIEF}_{G_2}(v, C) = \mathrm{NIEF}_G(v, C)$. Lastly for the P score, we have that $\mathrm{P}(v, G_1) < \mathrm{P}(v, G)$, but the relationship between $\mathrm{P}(v, G)$ and $\mathrm{P}(v, G_2)$ is more complicated. Indeed, if $\mathrm{P}(v, G) = 1 - \epsilon$ for $\epsilon \ll 1$ then $\mathrm{P}(v, G_2) > \mathrm{P}(v, G)$, whereas if $\mathrm{P}(v, G) = \epsilon \ll 1$ then $\mathrm{P}(v, G_2) < \mathrm{P}(v, G)$, and this is a consequence of the law of large numbers. To summarize, $\mathrm{IEF}(v, C)$ is invariant under transform 1 which increases $\mathrm{vol}(C)$ and under transform 2 which doubles the number of edges, $\mathrm{NIEF}(v, C)$ changes under transform 1 but not 2, and $\mathrm{P}(v, C)$ changes under both transforms.

Let us continue with two examples showing this result in action.

Example 1. Let $G = (V, E)$ be a graph with $\mathrm{vol}(V) = 10,000$, let $C_1, C_2 \subset V$ be communities in G with $\mathrm{vol}(C_1) = \mathrm{vol}(V)/2 = 5,000$ and $\mathrm{vol}(C_2) = 100$, and let $v \in V$ be a node with $\deg(v) = 5$ such that all edges from v connect to either C_1 or C_2. Table 1 shows the three CAS scores as $\deg_{C_1}(v)$ and $\deg_{C_2}(v)$ vary, highlighting the point in each score where the association strength of v into C_2 becomes larger than that of C_1.

We find that $\mathrm{IEF}(v, C_1) < \mathrm{IEF}(v, C_2)$ precisely when $\deg_{C_1}(v) < \deg_{C_2}(v)$. However, for both NIEF and P, there is a large penalty associated with C_1 compared to a negligible penalty associated with C_2. Note that in this case, the bias towards the smaller community is stronger for P than for NIEF. In our testing, we found this trend to be true in general.

$\deg_{C_1}$	$\deg_{C_2}$	IEF(C_1)	IEF(C_2)	NIEF(C_1)	NIEF(C_2)	P(C_1)	P(C_2)
5	0	1	0	0.5	0	0.97	0
4	1	0.8	0.2	0.3	0.19	0.81	0.95
3	2	0.6	0.4	0.1	0.39	0.5	1
2	3	0.4	0.6	0	0.59	0.19	1
1	4	0.2	0.8	0	0.79	0.03	1
0	5	0.0	1.0	0	0.99	0	1

Table 1. A comparison of CAS scores for 2 communities $C_1, C_2 \subset V$ with $\mathrm{vol}(V) = 10,000$, $\mathrm{vol}(C_1) = 5,000$ and $\mathrm{vol}(C_2) = 100$. The scores are rounded to 2 decimal places, and node v is omitted from the notation. Here, $\deg(v) = 5$ and each row represents a different split of $\deg(v)$ into C_1 and C_2. The grey cells highlight when a CAS score favours C_2 over C_1.

Example 2. Consider a graph $G = (V, E)$, a community C with $\mathrm{vol}(C) = \mathrm{vol}(V)/2$, and a node $v \in V$. Table 2 shows NIEF and P scores for (v, C) as $\deg(v)$ and $\deg_C(v)$ vary whilst $\deg_C(v)/\deg(v)$ remains constant.

The increasing values of $\mathrm{P}(v, C)$ as $\deg(v)$ and $\deg_C(v)$ increase is a consequence of the law of large numbers for the binomial distribution. Ultimately, what this example shows is that the P score, conditioned on $\deg_C(v)/\deg(v) > w(C)$, is biased towards nodes with a larger degree, whereas the NIEF score has no such bias.

deg	$\deg_C$	NIEF(C)	P(C)
3	2	0.17	0.5
6	4	0.17	0.66
9	6	0.17	0.75
12	8	0.17	0.81
15	10	0.17	0.85

Table 2. The comparison of 2 CAS scores on community C with $\mathrm{vol}(C) = \mathrm{vol}(V)/2$. The scores are rounded to 2 decimal places, and the node v is omitted from the notation.

2.3 The quality of CAS scores

To test the quality of the various community association strength scores, we need graphs with ground-truth communities, where the communities do not necessarily form a partition of the nodes. To this end, we will test a family of synthetic models stemming from the Artificial Benchmark for Community Detection (ABCD) model. The ABCD model [13] is a synthetic model with ground-truth communities and has 8 parameters to control the number of nodes, the degree and community size distributions, and the fraction ξ of noise; $1-\xi$ is (roughly) the fraction of edges of a node into its community. A fast, multi-threaded implementation of ABCD (ABCDe) was introduced in [17], and a hypergraph generalization (h-ABCD) was introduced in [16]. Importantly for our research, the ABCD model was generalized (ABCD+o) to include outliers [15], and further generalized (ABCD+o^2) to allow for overlapping communities. The latter model includes a parameter $\eta \geq 1$ which governs the expected number of community memberships for non-outlier nodes.

We first test the CAS scores' ability to rank communities based on the likelihood of a node being in those communities. For a graph G, a collection of ground-truth communities $\mathcal{C}$, and a CAS score $f : V \times 2^V \to [0,1]$, let $V_k \subseteq V$ be the set of nodes such that $v \in V_k$ if and only if v is contained in at least k communities in $\mathcal{C}$. For each score f, each relevant k, and each $v \in V_k$, we consider the k^{th} highest ranking community in $\mathcal{C}$ according to f and check if this community indeed contains v. Figure 1 presents the experiment results using ABCD+o^2 graphs with two different noise parameters: $\xi = 0.35$ and $\xi = 0.65$. All three measures perform similarly, with NIEF and P performing slightly better. The results suggest that each of the measures can accurately predict 1 or 2 communities a node is a member of, and with a low noise parameter, the prediction accuracy remains high as the number of communities increases.

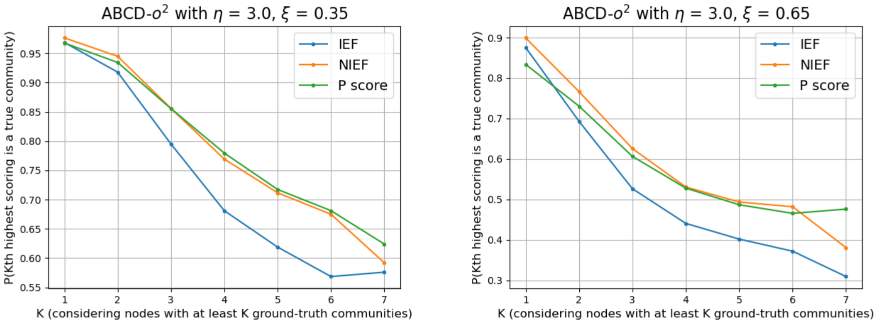

Fig. 1. Proportion of K^{th} highest scoring community that are actually ground-truth communities for the three CAS scores. Results are averaged over 10 ABCD+o^2 graphs with 10,000 nodes including 250 outliers, overlap parameter $\eta = 3$ and noise parameters $\xi = 0.35$ (left) and $\xi = 0.65$ (right). We can see that with all scores, the first few highest-scoring communities are almost always ground-truth, and this slowly degrades as K increases, with the NIEF and P scores decreasing more slowly.

Next, we test the scores' ability to distinguish outliers from non-outliers. Let $V_o \subseteq V$ be the outlier nodes concerning ground-truth communities $\mathcal{C}$. For each CAS score f, we order the nodes in V from smallest to largest based on $\max\{f(v,\cdot)\}$. We predict that, for $u, v \in V$ with $\max\{f(u,\cdot)\} < \max\{f(v,\cdot)\}$, u is more likely to be an outlier than v. We then compare our prediction to the ground truth with a receiver operating characteristic (ROC) curve. Figure 2 summarizes the results of this experiment by showing ROC curves for ABCD+o^2 graphs with 10,000 nodes (including 250 outliers), moderate noise ($\xi = 0.55$) and two different values for η. We also show each score's area under the ROC curves (AUC). While all measures get almost perfect results when $\eta = 1$ (no community overlap), we see degradation in the presence of community overlap ($\eta = 3$). We again see that all three measures perform similarly, with NIEF performing slightly better.

These two experiments indicate that CAS scores can capture local properties useful to refine node partitions. In the next section, we consider more realistic experiments where we do not explicitly use the ground-truth communities (but we use them for evaluation).

3 Using CAS scores in practice

In the previous section, we tested the quality of CAS scores with respect to the ground-truth communities of a graph. In practice, however, we may not have access to the ground-truth communities and would thus like to leverage CAS scores to help recover said communities. To this end, we present three scenarios where CAS scores can be useful. Section 3.1 describes an improvement to an existing consensus clustering algorithm. We then show applications of

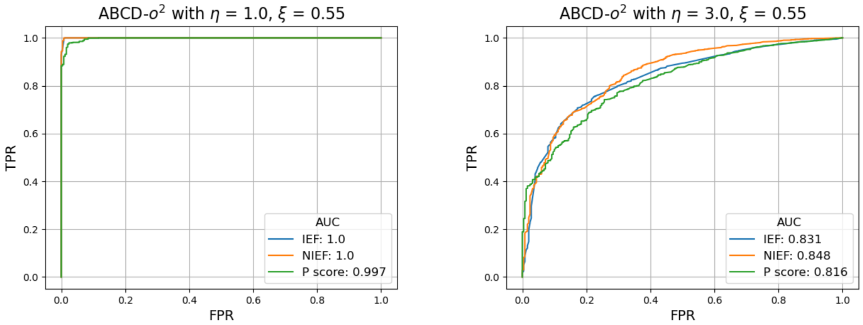

Fig. 2. ROC curves and corresponding AUC measures for three CAS scores for ABCD+o^2 graphs.

CAS scores to post-process any partitioning algorithm: identifying outlier nodes in Section 3.2 and identifying multi-community memberships in Section 3.3. Finally, we apply all three ideas to the well known college football graph in Section 3.4.

3.1 Improving graph partitions

In this section, we use CAS scores to modify an existing clustering algorithm called Ensemble Clustering for Graphs (ECG) [24,25] which is itself a modification of the Louvain algorithm. Let us briefly describe the ECG, and then our modification to ECG (which we will refer to as CAS-ECG).

The ECG algorithm takes a graph $G = (V, E)$ and a positive integer k as input and executes as follows.

1. Perform the first iteration of the standard Louvain algorithm on G, k times, to yield k "level 1" partitions of V.
2. Weight E such that the weight of an edge (u, v) is the fraction of partitions from step 1 with u and v in the same part.
3. Run Louvain, Leiden or some other partitioning algorithm on this weighted graph.

The full description of the algorithm can be found in Section 3 of [24].

We now detail CAS-ECG. During step 2 of ECG, edge (u, v) is weighted by how often u and v end up in the same part after 1 step of Louvain. Given some partition $\mathcal{C}$ of V with $u \in C_u$ and $v \in C_v$, write

$$\operatorname{ecg}\big((u,v),\mathcal{C}\big) = \begin{cases} 1 & C_u = C_v \\ 0 & C_u \neq C_v \end{cases}$$

Then the weight assigned to (u, v) at the end of step 2 of the ECG algorithm is $\sum \operatorname{ecg}\big((u,v),\mathcal{C}\big)$ where the sum is taken over the k partitions. We consider replacing this weight with a new weight based on a given CAS score. Intuitively, we want (u, v) to receive a higher weight if u is strongly associated with C_v and/or v is strongly associated with C_u, even when $C_u \neq C_v$. For a CAS scoring function f, we propose the following two options for weighting the edges.

$$\begin{aligned} f_{or}\big((u,v),\mathcal{C}\big) &= f(u,C_v) + f(v,C_u) - f(u,C_v)\cdot f(v,C_u)\,. \\ f_{and}\big((u,v),\mathcal{C}\big) &= f(u,C_v)\cdot f(v,C_u). \end{aligned}$$

We aim to find a CAS score f and a weighting scheme f_{or} or f_{and} that improves ECG. Thus, we test the performance of the six combinations of scores and weighting schemes to see if any can improve ECG's ability to recover ground-truth communities. We perform this test on ABCD graphs with 10,000 nodes, a minimum degree of 5, a minimum community size of 50, and varying levels of noise. Each clustering uses $k = 16$ runs in the ensemble step and obtains the final clustering from running the Leiden algorithm on the weighted graph. The results of this experiment are presented in Figure 3. We find that each of the six modifications provides comparable results to the base ECG algorithm for $\xi < 0.55$. Although five out of the six configurations seem to yield comparable or worse results than ECG, we find that using P_{and} as the edge weighting function results in a substantial increase in the Adjusted Mutual Information (AMI) score between noise levels $\xi = 0.55$ and $\xi = 0.65$, with a peak increase of about 5%. As can be seen in the left plot, this range of noise levels corresponds to the "critical" region where improvements are significant; with lower noise values, most algorithms will yield good results, while for higher noise values, the resulting AMI scores are very low no matter which algorithm is used.

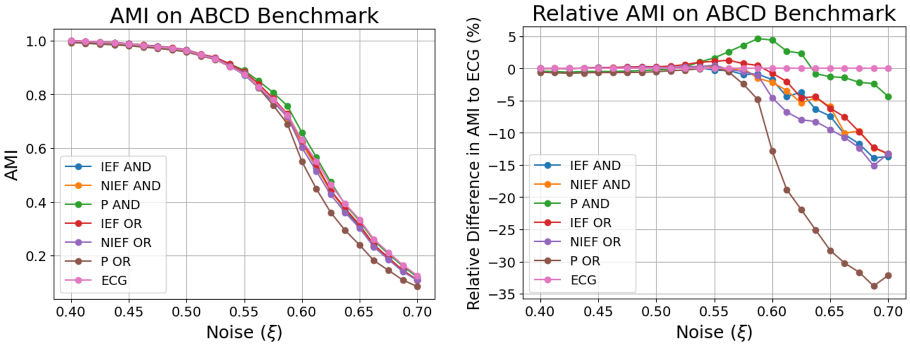

Fig. 3. (Left) Average AMI using each CAS and weight functions and the average relative change compared to the base ECG method (Right). Any option performs similarly to ECG, with P_{and} performing slightly better on graphs with ξ between 0.55 and 0.65. For each value of ξ, 50 ABCD graphs were generated.

3.2 Outlier detection

This section uses a CAS score to post-process a partitioning algorithm to find outliers. Let $G = (V, E)$ be a graph with ground-truth communities and a set of outliers $V_o \subseteq V$, and let $\mathcal{C}$ be a partition of V found by a detection algorithm. Then, although $\mathcal{C}$ is not the ground truth assuming $|V_o| > 0$, we can still attempt to recover as much of V_o as possible by finding nodes v such that $\text{CAS}(v, C) \approx 0$ for all $C \in \mathcal{C}$. To test if this heuristic is feasible, we perform the following experiment on the ABCD+o model using each of the 3 CAS scores. First, we use the Leiden algorithm to obtain a partition of the nodes. We then use the CAS score of a node and its suggested community to rank the nodes from most likely to be an outlier (low maximum CAS score) to least likely to be an outlier (high maximum CAS score). Finally, we compute the area under the ROC curve (AUC) for our outlier prediction. For this experiment, we fix $|V| = 10,000$, and we fix the distributions for sampling degrees and community sizes. We test noise parameter ξ varying from 0.45 to 0.7, and a number of outliers varying from 100 to $4,000$ (corresponding to %1 to %40 of the nodes). The results are shown in Figure 4 and are averaged over 50 independent graphs for each configuration. We observe decreasing performance as the noise level ξ increases and slightly decreasing performance as the number of outliers increases. Each measure performs similarly, with a slight advantage for the NIEF score. These results show that on ABCD+o graphs and using Leiden to find an initial partition, all 3 CAS scores can fairly accurately recover the outliers when $\xi \leq 0.5$, whereas none of the scores can recover the outliers when $\xi \geq 0.6$.

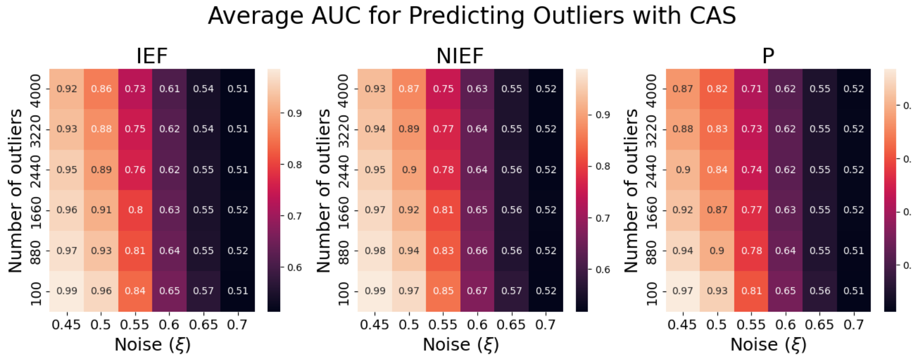

Fig. 4. Average AUC score using Leiden followed by CAS-based rankings on 50 ABCD-o graphs with 10,000 nodes.

3.3 Multi-communities

Finding overlapping communities in a graph is substantially more difficult than finding outliers, particularly when the graphs are noisy and/or the communities are numerous and small. It can quickly become difficult to distinguish noise edges from within-community edges. Here, we show how to use a CAS score to refine a collection of communities. Let $G = (V, E)$ be a graph, $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a collection of communities in G found by some algorithm, f be a CAS score, and $\tau > 0$ be a threshold. Construct a new collection of communities, $\mathcal{C}'$, as follows.

1. Initially, $\mathcal{C}' = \{C'_1, \ldots, C'_k\}$ is a collection of empty sets.
2. For all $v \in V$ and all $C_i \in \mathcal{C}$, if $f(v, C_i) \geq \tau$ then add v to C'_i in $\mathcal{C}'$.

To test this refinement process, we use the NIEF score to refine communities obtained via the ego-split method. In this method, a partition is found on an auxiliary graph G'. This auxiliary graph contains one or more copy of each node from the original graph G, depending on the community structure of each node's ego-net. The partition is then mapped back to G, yielding overlapping communities (more details can be found in [6] and [14]). We consider ABCD+o^2 graphs with 10,000 nodes, a fixed noise level $\xi = 0.35$, and varying η values. The results are presented in Figure 5. To measure the quality of predicted communities compared to the ground truth, we use the overlapping Normalized Mutual Information (oNMI) measure: a similarity measure for two collections of subsets $\mathcal{X}, \mathcal{Y}$ of a set S [21]. For each $\eta \in \{1, 1.5, 2, 2.5, 3\}$, we first find $\mathcal{C}_{guess}$ via the ego-split method (restricting the minimum community size to 10) and compute oNMI$(\mathcal{C}_{guess}, \mathcal{C}_{true})$. We then obtain $\mathcal{C}'_{guess}$ using our refinement method with a variety of thresholds τ and compare the resulting oNMI scores with the original. We find that, while no single value for τ is clearly the best, choosing $\tau \in [0.075, 0.25]$ yields a refinement $\mathcal{C}'_{guess}$ that is better than the initial prediction $\mathcal{C}_{guess}$. This result suggests that, with a well-chosen CAS score and threshold, the refinement process can indeed improve existing detection algorithms.

In practical applications, since the ground truth is unknown, it is not always clear how to pick a good threshold for the refinement process. We propose a guiding method in Figure 6 where we show the number of outliers obtained for each choice of threshold. The true number of outlier nodes is 250 (shown with a dashed line). While this information is also not likely known beforehand, it gives us some rule of thumb to set the threshold. For example, if we suspect the number of outliers to be small (as it is), then a threshold around 0.1 for NIEF seems like a good choice, possibly slightly higher if we suspect no or little overlap ($\eta \approx 1$) and slightly lower if we suspect lots of overlap. This threshold corresponds to good results in Figure 5.

3.4 Illustration on a real graph

We now illustrate the methods described in Sections 3.1, 3.2, and 3.3 using a real-world graph. We consider the college football graph from [8] with corrections

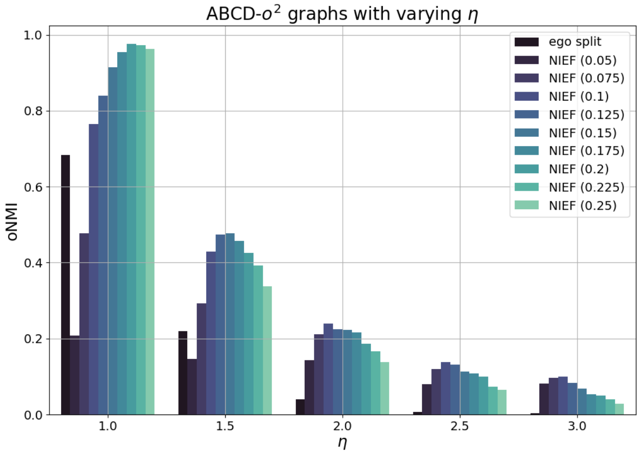

Fig. 5. Ego-splitting (darker bars) followed by NIEF with varying threshold values for ABCD+o^2 graphs with 10,000 nodes, $\xi = 0.35$, and varying η value. We compare each set of communities with the ground truth via the oNMI measure.

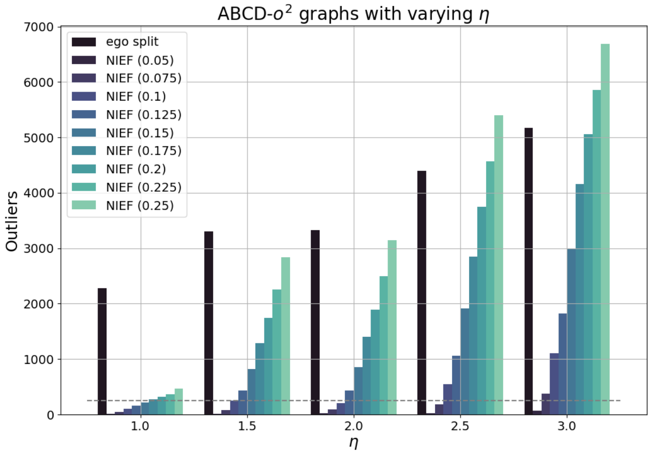

Fig. 6. Ego-splitting (darker bars) followed by NIEF with varying threshold values for ABCD+o^2 graphs with 10,000 nodes, $\xi = 0.35$, and varying η values. We compare the number of outlier nodes produced in each case.

to the labels as described in [7]. The graph has 115 nodes (teams) and 613 edges (games played). After the corrections, there are 12 communities corresponding to football conferences. In general, teams play most games within their conference.

One of these communities is in fact a group of independent teams which we use as a surrogate for outlier nodes.

We first consider the CAS-ECG algorithm from Section 3.1. In Figure 7, we show the graph using a forced directed layout[5], where the node colours correspond to the different conferences, and the outlier nodes are shown as black triangles (left plot). Running the CAS-ECG algorithm using the P_{and} weighting scheme, the same force-directed layout algorithm yields the (much nicer) plot on the right. Moreover, we get slightly better communities with CAS-ECG than with Leiden, i.e. larger AMI values.

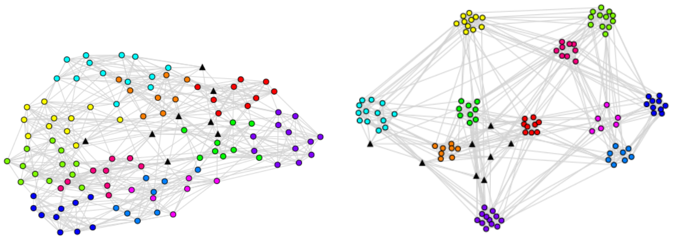

Fig. 7. Football graph displayed using a force-directed layout algorithm. The left plot has all unit edge weights, while the right plot has edge weights derived from the CAS-ECG algorithm. Communities are shown in colors, and outlier nodes as black triangles.

Next, we use the P score to rank nodes as possible outliers as per Section 3.2. The results are presented in Figure 8. We run two versions of the experiment, the first using Leiden to obtain the initial partition and the second using the P_{and} version of CAS-ECG. We see that the eight outlier nodes are found in the top 10 (Leiden) or the top 8 (CAS-ECG). Moreover, the CAS-ECG algorithm slightly outperforms Leiden in recovering the ground-truth communities.

Finally, while there is no clear community overlap in this dataset, we look at the most likely node(s) that are part of multiple communities, again using the P_{and} version of CAS-ECG. In Figure 9, we show the ego-nets for two nodes with high P scores for two communities. The nodes are shown as larger circles. In both cases, while the nodes are both part of a tight community (magenta), they seem to act as a bridge to other communities.

4 Conclusion

We presented three community association strength functions, highlighted their similarities and differences, and showed their ability to recover community involvement in a network. We suggested multiple ways to leverage community association strength when detecting communities with overlap and outliers. While

[5] the Fruchterman-Reingold algorithm in Python-igraph

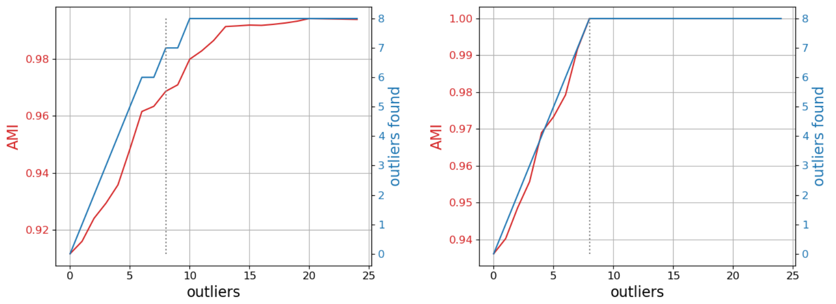

Fig. 8. Outlier detection on the football graph respectively using Leiden (left) and the P_{and} modified ECG (right), followed by node ranking via P score. We show the number of outliers found (blue curves) and the corresponding AMI (red curves) as we iterate through the ranked list.

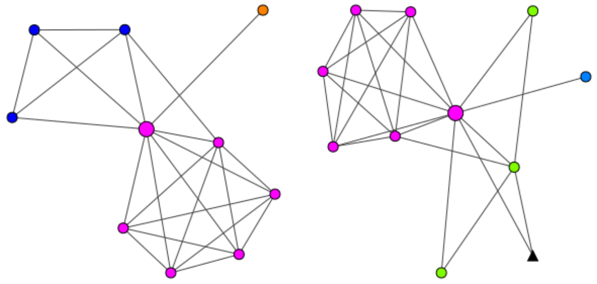

Fig. 9. Ego-nets for two nodes (shown with larger circles) having large CAS P-scores toward two communities.

the experiments in Section 3 suggest that CAS scores do a good job at improving community detection, our goal is merely to show that improvement is *possible*. On the one hand, our experiments should be tested on a wide variety of real datasets with ground-truth communities to obtain more conclusive results. On the other hand, there might be stronger CAS scores and/or more clever uses of these scores that outperform what we have presented here. In an upcoming journal version, we will present more experiments on more datasets and delve deeper into the meaning behind the results. Furthermore, we leave it as an open problem to find CAS scores that can outperform the three scores presented here.

References

1. Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.
2. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

3. Kuijjer ML. Calderer G. Community detection in large-scale bipartite biological networks. *Frontiers in Genetics*, 12, 2021. doi:10.3389/fgene.2021.649440.
4. Fan Chung Graham and Linyuan Lu. *Complex graphs and networks*. Number 107 in CBMS Regional Conference Series in Mathematics. American Mathematical Soc., 2006.
5. Imre Derényi, Gergely Palla, and Tamás Vicsek. Clique percolation in random networks. *Physical review letters*, 94(16):160202, 2005.
6. Alessandro Epasto, Silvio Lattanzi, and Renato Paes Leme. Ego-splitting framework: From non-overlapping to overlapping clusters. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 145–154, 2017.
7. Tim S Evans. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(12):P12037, 2010.
8. Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
9. Zx. Han, Ll. Shi, and L. et al. Liu. H-Louvain: Hierarchical Louvain-based community detection in social media data streams. *Peer-to-Peer Networking and Applications*, 17, 2024. doi:10.1007/s12083-024-01689-9.
10. Akhil Jakatdar, Baqiao Liu, Tandy Warnow, and George Chacko. AOC: Assembling overlapping communities. *Quantitative Science Studies*, 3(4):1079–1096, 2022.
11. Bogumił Kamiński, Bartosz Pankratz, Paweł Prałat, and François Théberge. Modularity of the ABCD random graph model with community structure. *Journal of Complex Networks*, 10(6):cnac050, 2022. doi:10.1093/comnet/cnac050.
12. Bogumił Kamiński, Paweł Prałat, François Théberge, and Sebastian Zajac. Predicting properties of nodes via community-aware features. *Social Network Analysis and Mining*, 14(1):117, 2024. doi:10.1007/s13278-024-01281-2.
13. Bogumił Kamiński, Paweł Prałat, and François Théberge. Artificial benchmark for community detection (ABCD)—fast random graph model with community structure. *Network Science*, pages 1–26, 2021.
14. Bogumił Kamiński, Paweł Prałat, and François Théberge. *Mining Complex Networks*. Chapman and Hall/CRC, 2021.
15. Bogumił Kamiński, Paweł Prałat, and François Théberge. Artificial benchmark for community detection with outliers (abcd+o). *Applied Network Science*, 8(1):25, 2023.
16. Bogumił Kamiński, Paweł Prałat, and François Théberge. Hypergraph artificial benchmark for community detection (h-abcd). *Journal of Complex Networks*, 11(4):cnad028, 2023.
17. Bogumił Kamiński, Tomasz Olczak, Bartosz Pankratz, Paweł Prałat, and François Théberge. Properties and performance of the abcde random graph model with community structure. *Big Data Research*, 30:100348, 2022. URL: https://www.sciencedirect.com/science/article/pii/S2214579622000429, doi:10.1016/j.bdr.2022.100348.
18. Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.
19. Andrea Lancichinetti, Filippo Radicchi, José J. Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 2011. doi:10.1371/journal.pone.0018961.

20. Borut Lužar, Zoran Levnajić, Janez Povh, and Matjaž Perc. Community structure and the evolution of interdisciplinarity in Slovenia's scientific collaboration network. *PLOS ONE*, 9(4):1–5, 04 2014. `doi:10.1371/journal.pone.0094429`.
21. Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011.
22. Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
23. Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24:515–554, 2012. URL: `https://api.semanticscholar.org/CorpusID:15719130`.
24. Valérie Poulin and François Théberge. Ensemble clustering for graphs. In *International Conference on Complex Networks and their Applications*, pages 231–243. Springer, 2018.
25. Valérie Poulin and François Théberge. Ensemble clustering for graphs: comparisons and applications. *Applied Network Science*, 4(4), 2019. `doi:10.1007/s41109-019-0162-z`.
26. S. Rahiminejad, M.R. Maurya, and S. Subramaniam. Topological and functional comparison of community detection algorithms in biological networks. *BMC Bioinformatics*, 20, 2019. `doi:10.1186/s12859-019-2746-0`.
27. Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.