IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 13, 2022, accepted March 28, 2022, date of publication April 13, 2022, date of current version April 19, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3167060

A Modular Framework for Centrality and Clustering in Complex Networks

FRÉDÉRIQUE OGGIER[1], SILIVANXAY PHETSOUVANH[2], AND ANWITAMAN DATTA[3]
[1]Division of Mathematical Sciences, Nanyang Technological University, Singapore 639798
[2]Ministry of Technology and Communications, Vientiane 01000, Laos
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Anwitaman Datta (anwitaman@ntu.edu.sg)

The work of Frédérique Oggier was supported by Nanyang Technological University (NTU), Singapore, under the Start-Up Grant. The work of Silivanxay Phetsouvanh was supported by the Ph.D. Scholarship through NTU funded by the Ministry of Education, Singapore.

ABSTRACT The structure of many complex networks includes edge directionality and weights on top of their topology. Network analysis that can seamlessly consider combination of these properties are desirable. In this paper, we study two important such network analysis techniques, namely, centrality and clustering. An information-flow based model is adopted for clustering, which itself builds upon an information theoretic measure for computing centrality. Our principal contributions include (1) a generalized model of Markov entropic centrality with the flexibility to tune the importance of node degrees, edge weights and directions, with a closed-form asymptotic analysis, which (2) leads to a novel two-stage graph clustering algorithm. The centrality analysis helps reason about the suitability of our approach to cluster a given graph, and determine 'query' nodes, around which to explore local community structures, leading to an agglomerative clustering mechanism. Our clustering algorithm naturally inherits the flexibility to accommodate edge directionality, as well as different interpretations and interplay between edge weights and node degrees. Extensive benchmarking experiments are provided, using both real-world networks with ground truth and synthetic networks.

INDEX TERMS Directed weighted graphs, entropy, centrality, graph clustering, random walkers.

I. INTRODUCTION

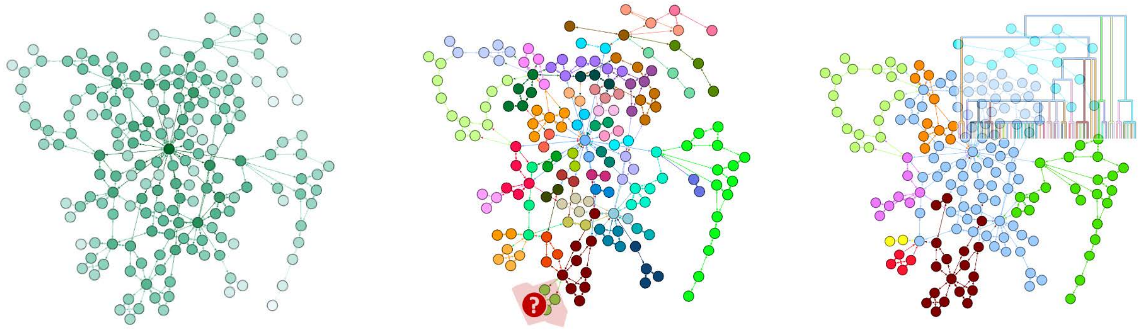
Centrality, a measure of node importance, and clustering (also known as community detection) are two popular techniques used to study the structure of complex networks.

The goal of this paper is to study these with the flexibility to modulate the directionality and weights of the edges, capturing a spectrum of behaviors for a given complex network. A motivating example is the Bitcoin network, where individuals or nodes are Bitcoin wallet addresses, and edges represent transactions from one wallet to another: we may want to identify users interacting with each other, or further distinguish senders from receivers, or take into account the transaction amounts, each of these combinations providing different perspectives on the same network.

To do so, we focus on entropy-based centralities [31], [48]. They are flow-based measures [6], well suited for studying phenomena such as the movement of funds, e.g., Bitcoins, among users, where the volume of the flow is conserved. We analyze a particular such centrality, which we refer as Markov entropic centrality, which we then leverage to propose a versatile flow circulation based graph clustering algorithm, relying on a random walker [1], [4], [20]. The algorithm's basic premise is that random walks originating at a given 'query' node, which we characterize in terms of its Markov entropic centrality, is more likely to stay confined within a subgraph which forms a local community containing the query node, than to visit nodes outside said community. The probability distribution of a random walker's visit (and absorption) at different nodes in the graph is used as a signal to identify cluster boundaries.

The first part of the paper is dedicated to the analysis of the proposed Markov entropic centrality, respectively for unweighted (Section II) and weighted (Section III) graphs. We characterize it as 'Markov', as opposed to the original path based entropic centrality [48], because our model leverages the random walks being absorbed in an abstraction called auxiliary nodes with a certain absorption probability, and the future steps of the random walker depends solely on its current location, irrespective of its past. This corresponds to

The associate editor coordinating the review of this manuscript and approving it for publication was Hocine Cherifi.



Step 1: For a random walk starting at each node, determine the probability with which it terminates at every node. Quantify this uncertainty through entropy to determine each node's entropic centrality.

Step 2: Consider low-centrality nodes as query nodes ?, check the probability distributions of random walks from said nodes, as determined in Step 1, to identify sharp changes in probabilities, to identify their local communities. Continue the process to cover the whole graph.

Step 3: Create a meta-graph with its nodes as the local clusters from step 2, and repeat Steps 1 and 2 on this graph in order to obtain larger communities using an agglomerative hierarchical approach.

FIGURE 1. A flowchart of the steps at a high level for agglomerative clustering leveraging entropic centrality.

Step 1 in Figure 1, where we provide the big picture of the overall flow of steps for the exploration of node centralities and its application to carry out clustering in an agglomerative manner. In Section II, we analyze the behavior of the Markov entropic centrality for unweighted graphs as a function of (a) the absorption probability that the walker stops at given nodes, and (b) time (or number of steps in the walk). Results are analytically proven in terms of bounds and a closed form expression captures the asymptotic behavior of the random walk probability distribution. We explore as particular cases when the absorption probability is either a constant at all nodes, or is dependent on the individual nodes' degrees — which in turn makes the model versatile to capture different situations. This theory is generalized to weighted graphs in Section III, relying on the concept of weighted entropy to appropriately interpret edge weights. Section IV contains the proposed clustering (community detection) algorithm. The above random walker probability distribution analysis directly lends itself to the design of a novel graph clustering algorithm. Foremost, we observe that: (i) nodes with a low entropic centrality have local clusters where ties are relatively strong, while nodes tend to have a relatively high entropic centrality when they either are embedded within a large cluster, or when they are at the boundary of two clusters (and could arguably belong to either), and moreover, (ii) sharp changes in the random walker absorption probability distribution signal boundaries of local community structures. This leads to a two-stage clustering algorithm (corresponding to Steps 2 and 3 in Figure 1): First, observing that nodes at the boundary of clusters act as bridges and have high centrality, we initiate the exploration of meaningful node centric local communities around nodes with low entropic centrality. This approach inherits the flexibility of the underlying centrality model, and is applicable to the spectrum of combinations (un/directed, un/weighted) of graphs. While we specifically and exclusively apply our own centrality model to determine the suitability of query nodes for exploring community structures, combining it with other complementary approaches to identify bridge nodes, e.g., [9], [30], and accordingly exclude them from being query nodes can be readily incorporated in our community detection framework. Second, the process is reapplied on the created clusters (instead of the nodes) to effectuate a bottom-up, scalable, hierarchical clustering. Working principles behind the heuristics are supported by formal derivations, and the performance of the algorithms has been determined with extensive experiments using real as well as synthetic networks.

Evaluating in general the quality of a clustering algorithm is a debated topic [38], [46], [49]. Complications arise because of multiple reasons, including that different clustering algorithms may identify communities which may reflect different aspects of relationships; there may be multiple interpretations of relationships, resulting in different 'ground truths' being valid. We thus report benchmarking from a variety of networks, falling into two categories: (a) synthetic networks [22], [23], ranging from small graphs (of 100 to 500 nodes) to larger graphs (of 1000 and 4000 nodes), for which the proposed algorithm fares well and is comparable with the standard Louvain [5], and (b) real world graphs with meta-information used to assign a ground truth; these include the classical karate club [51] and the small cocaine dealing network [11] to explain our model and validate the corresponding findings, but also the dolphin network [28] and the American football network [15], for which the performance of the proposed clustering algorithm is compared with that of Louvain [5], Infomap [44] and label propagation [2].

Furthermore, clustering of Bitcoin transaction subgraphs [32], [33] is performed to explore the versatility of our model vis-à-vis the spectrum of choices in interpreting the directionality and weights of edges. Since no ground truth is known, we compare the obtained results with an existing Bitcoin forensics study [41].

Remark 1: All experiments and implementations were done in Python, using NetworkX [19] and Scikit-Learn [37]. All graphs were drawn with either NetworkX or Gephi [3].

Remark 2: This work builds upon preliminary ideas explored in Chapter 4 of the doctoral dissertation [40] of the second author. In contrast to the treatment of the thesis, where the heuristics are presented and evaluated with experiments, this article includes rigorous formal analysis that helps understand the underlying principles and thus the rationale of the design of the algorithms. This is accompanied with additional experiments to validate the analysis as well as benchmark the algorithms more exhaustively.

A. RELATED WORKS

Entropic centrality was introduced in [48], to capture the idea of path-transfer flow. It was then extended to non-atomic flows in [34], [35] and to a Markov model in [31] by relaxing the assumption (in [48]) that a random walker cannot revisit previously traversed nodes. Some of the limitations of [31], such as the lack of analysis of the centrality measure and no variation for weighted graphs, serve, in part, as motivations for this work. We provide, in contrast, a closed form analysis, which leads to both a precise understanding, and computational efficiency and scalability of the model. A definition is proposed for weighted graphs, which is consistent with the known notion of weighted entropy.

There is a body of work on community-aware centrality measures, e.g, [13], [14], [18], [21], [27], [43], [47], [52]. These studies span across overlapping and non-overlapping communities, and determine the most influential nodes. This in turn has been used to design more effective information dissemination or immunization strategies in several of the above mentioned works (see [10] for an overview). Our work fits within this general premise of considering an underlying coupling between community structures and node centrality. However the emphasis of our work is to demonstrate how one can leverage it in a complementary manner, in identifying the originally implicit community structures themselves using such a community-aware centrality measure.

¯A notion of weighted degree centrality has been proposed [36], which also contains an elaborate discussion on interpreting centrality for weighted graphs. Motivated by these considerations, we propose a notion of flow-based centrality which is seamlessly adjustable to undirected, directed and/or weighted graphs. This is in contrast to [31], where weighted graphs are dealt with by changing the transition probabilities of the random walker as per edge weights, which is one specific instance within our tunable model.

In [45], two broad families of undirected graph clustering methods have been identified - (i) those based on vertex similarity, e.g., distance or similarity measure, adjacency or connectivity based measures; and (ii) those based on cluster fitness measures, including density and cut-based measures, e.g., modularity. In [29], the former are referred to as pattern based, since these methods go beyond basic edge density characteristics. They include algorithms relying on random walkers [1], [4], [20]. Most algorithms for directed graphs [29] rely on creating an ‘‘undirected’’ representation of the directed graph considered, which may result in the loss of semantics and information contained in the directionalities. Examples of notable exceptions where the clustering algorithm is designed specifically for directed graphs are the information flow approaches of [44], where clusters are identified as subgraphs where the information flow within is larger compared to outside node groups, and [20], where an information flow-based cluster is a group of nodes where a random walker (referred as surfer) is more likely to get trapped, rather than moving outside the group. Some random walker based approaches [1], [4], because of the additional constraints that they impose on the walkers, are only suitable for undirected unweighted graphs, unlike our proposed algorithm. Furthermore, the strong coupling of our algorithm's design with the Markov entropic centrality allows to initiate the random walker in an informed manner, in contrast to [1], [4], which, in absence of any guidance on where to initiate random walks, require more complex constrained random walkers.

In [31], a radically different approach (adapting [15]) for community detection is applied, where edges are iteratively removed, and in each iteration one needs to identify the edge such that the average entropic centrality over all the nodes is reduced the most. This requires the computation of entropic centrality of all the nodes multiple times for the different resulting graph snapshots from edge removals, and do so numerically over and over again. Naturally, the approach is computationally intensive and not scalable. For a specific undirected unweighted graph, our algorithm took 1.072 seconds to compute the communities, which is multiple orders of magnitude faster than the 3196.07 seconds needed by the edge removal algorithm [31] (more details on the specific experiments can be found in Subsection IV-D). The communities detected by our approach were also qualitatively better for the network where the two approaches were compared head-to-head. Moreover, the authors in [31] noted that their approach was not useful for community detection in directed graphs. In contrast, we demonstrate with experiments, that our approach works also with directed graphs.

B. CONTRIBUTIONS

Our first set of contributions is vis-à-vis entropic centrality. We provide (i) a generalization of the entropic centrality, which can be tuned to capture a spectrum of combinations in terms of the role of the edges' weights and directionality, accompanied with (ii) a mathematically rigorous analysis to understand the role of the model parameters and choose them judiciously, and ultimately resulting in (iii) a computationally, significantly efficient and thus scalable model with respect to prior work.

The second set of contributions is vis-à-vis graph clustering. The salient aspects of our work comprise the use of entropic centrality (and underlying analysis) to (i) inform on whether our approach would yield good quality of clusters given a graph, and if so, (ii) how to identify good candidate 'query nodes' around which initial sets of local community structures should be explored, which in turn yields (iii) an effective novel two-stage clustering algorithm even while using a single random walker. Our clustering algorithm (iv) amortizes the previous computation of entropic centralities in computing these communities, and (v) inherits (from the underlying centrality model) the flexibility of interpreting the roles of edge weights and directionality, allowing for capturing different behaviors of a weighted directed graph as per application needs. Our approach is thus one of the few graph clustering approaches in the literature that handles directed graphs naturally, and additionally, it allows for flexible incorporation and interpretation of edge weights.

II. ENTROPIC CENTRALITY FOR UNWEIGHTED GRAPHS

Consider a connected directed graph $G = (V, E)$ with $n = |V|$ nodes labeled from 1 to n and $|E|$ directed edges, and a random walker on G, which starts a random walk at any given node according to an initial probability distribution. If the walker decides on which node v to walk to, based only on the current node u at which it currently is (v does not have to be distinct from u), the random walk is a Markov chain, which can be modeled by a stochastic matrix P where P_{uv} is the probability to go to node v from node u, and in general, P^k gives the transition probability of going from any state to any other state in k steps, $k \geq 1$. We will assume that every node has a self-loop (a self-loop is needed for the model to encapsulate the case of nodes with zero out-degree.) Let $d_{out}(u)$ denote the out-degree of u, which includes the self-loop as per our model. A typical choice that we consider in this section is $P_{uv} = \frac{1}{d_{out}(u)}$ for every node v belonging to the set $\mathcal{N}_u$ of out-neighbors of u (this is saying that each neighbor is chosen uniformly at random). Other choices are possible, and indeed, subsequently in this paper, we shall consider the case of weighted graphs where the transition probabilities depend on the edge weights.

A. MARKOV ENTROPIC CENTRALITY

Now, for every node v in G, an auxiliary node v' is added [31], together with a single directed edge (v, v'), and a probability $p_{vv'}$ to be chosen (the original probabilities p_{uv} are adjusted to the probabilities $\tilde{p}_{uv}$ so that the overall matrix remains stochastic). Once the flow reaches an auxiliary node, it is absorbed because the auxiliary nodes have no outgoing edges. This gives a notion of Markov entropic centrality as defined in [31].

Definition 1: The *Markov entropic centrality* of a node u at time t is defined to be:

$$C_H^t(u) = -\sum_{v \in V} (\tilde{p}_{uv}^{(t)} + p_{uv'}^{(t)}) \log_2(\tilde{p}_{uv}^{(t)} + p_{uv'}^{(t)}) \tag{1}$$

Here $\tilde{p}_{uv}^{(t)}$ denotes the probability to reach v (for any node v in V) from u at time t, where $p_{uv'}$ is the probability to reach an auxiliary node v' from u.

A node u is central if $C_H^{(t)}(u)$ is high: when $C_H^{(t)}(u)$ is high, using the underlying entropy interpretation, this means that for a random walker starting at node u, the uncertainty about its destination v after t steps is high, thus u is well connected.

The time parameter t used in the Markov model can also be interpreted as a notion of locality. It describes a horizon around the node u, of length t steps. Thus the entropic centrality at $t = 1$ emphasizes a node's degree, t being the graph diameter implies that we are considering a period of time by when the whole graph can be first reached, and $t \to \infty$ describes the asymptotic behavior over time. Thus $C_H^{(t)}(u)$ can be regarded as a measure of influence of u over its close neighborhood for small values of t, and over the whole graph asymptotically.

Given an $n \times n$ stochastic matrix P describing the moves of a random walker on a directed graph, let us then introduce $n = |V|$ auxiliary nodes, one for each node v, $v \in V$, with corresponding probabilities $p_{vv'} = D_{vv}$ to walk from node v to node v'. This means the out-degree of u increases by 1 because of the addition of u'. Nevertheless, by $d_{out}(u)$, we will actually refer to the degree of u before the addition of u'. This creates the following right stochastic matrix

$$\hat{P} = \begin{bmatrix} \tilde{P} & D \\ 0_n & I_n \end{bmatrix}$$

where $D = D_{uu}$ is a diagonal matrix, and $\tilde{P}$ is such that $[\tilde{P}, D]1 = 1$. We assume that $\tilde{P}$ has for u-th row $(\tilde{P}_{u,l})_l = (1 - p_{uu'})(P_{u,l})_l$. Then $\sum_{l=1}^{2n} (\hat{P})_{ul} = (1 - p_{uu'}) \sum_{l=1}^{n} p_{ul} + p_{uu'} = 1$ for every u. This is alternatively written as $\tilde{P} = (I_n - D)P$. The identity matrix I_n represents the stoppage of the flow at the auxiliary nodes (an absorption of the flow arriving at these nodes). To determine the centrality of a specific node u, we assume an initial distribution that gives a probability of 1 to start at u, and 0 elsewhere.

The definition of Markov entropic centrality was used as part of a clustering algorithm in [31], and the probabilities $p_{vv'}$ were explored numerically to optimize the clustering results. Our first contribution is the following closed-form expression for the asymptotic behavior of the transition matrix.

Lemma 1: For an integer $t \geq 1$ and $D \neq 0$, we have

$$\hat{P}^t = \begin{bmatrix} \tilde{P}^t & (\sum_{j=0}^{t-1} \tilde{P}^j) D \\ 0_n & I_n \end{bmatrix}. \tag{2}$$

In particular

$$\hat{P}^t = \begin{bmatrix} 0_n & (I_n - \tilde{P})^{-1} D \\ 0_n & I_n \end{bmatrix} \tag{3}$$

when $t \to \infty$.

Proof: Formula (2) follows from an immediate computation. Since $\tilde{P}$, D have non-negative real coefficients, so has $\tilde{P}^j D$, thus $(\sum_{j=0}^{l} \tilde{P}^j) D \geq (\sum_{j=0}^{l-1} \tilde{P}^j) D$ for any $l \geq 1$, and the equality holds if and only if $\tilde{P}^N = 0$ for some N. But this N

must exist when $t \to \infty$, because $\hat{P}$ is a stochastic matrix, meaning that the sum of every row must remain 1, while the coefficients of $(\sum_{j=0}^{t-1} \tilde{P}^j)D$ increases at every increment of $t \leq N$. Then

$$\hat{P}^N = \begin{bmatrix} 0_n & (\sum_{j=0}^{N-1} \tilde{P}^j)D \\ 0_n & I_n \end{bmatrix}.$$

However, since $(\tilde{P} - I_n)(\sum_{j=0}^{N-1} \tilde{P}^j) = \tilde{P}^N - I_n = -I_n$, the matrix $(\tilde{P} - I_n)$ is invertible and we have $\sum_{j=0}^{N-1} \tilde{P}^j = -(\tilde{P} - I_n)^{-1}$, yielding (3). ■

The matrix $\hat{P}^t$ in (2) contains a first block $\tilde{P}^t$ whose coefficients $\tilde{p}_{uv}^{(t)}$ are the probabilities to go from u to v in t steps. The second block $(\sum_{j=0}^{t-1} \tilde{P}^j)D$ contains as coefficients the probabilities to go from u to v' in t steps, which we denote by $p_{uv'}^{(t)}$. Therefore the probabilities $\tilde{p}_{uv}^{(t)} + p_{uv'}^{(t)}$ in (1) are obtained from the uth row of the matrix $\hat{P}^t$, by summing the coefficients in the columns v and v'. When $t \to \infty$, the term in column v becomes 0, and we are left with the term in column v', of the form

$$\pi_{uv} := p_{uv'}^{(\infty)} = \begin{cases} \sum_{t\geq 1} \tilde{p}_{uv}^{(t)} p_{vv'} & u \neq v \\ (\sum_{t\geq 1} \tilde{p}_{uv}^{(t)} + 1) p_{vv'} & u = v. \end{cases}$$

This means that we are looking at the probability to start at u and reach v in $t \geq 0$ steps, and then to get absorbed at v (that is reaching v', and then not leaving v'). Asymptotically, the entropic centrality defined in (1) becomes

$$C_H^\infty(u) = -\sum_{v\in V} \pi_{uv} \log_2(\pi_{uv}).$$

We discuss next how the choice of the probabilities $p_{uu'}$ to reach an auxiliary node u' from u influences the random walker at time t.

Lemma 2: The probability $\tilde{p}_{uv}^{(t)}$ to start a walk at u and to reach v in t steps is bounded as follows:

$$(1 - p_{uu'})(\max_{w\in V}(1 - p_{ww'}))^{t-1} p_{uv}^{(t)} \geq \tilde{p}_{uv}^{(t)} \geq (1 - p_{uu'})(\min_{w\in V}(1 - p_{ww'}))^{t-1} p_{uv}^{(t)}.$$

Proof: Since $\tilde{p}_{uv} = (1 - p_{uu'})p_{uv}$, we have

$$\begin{aligned} \tilde{p}_{uv}^{(2)} &= \sum_{w\in out(u)\cap in(v)} \tilde{p}_{uw}\tilde{p}_{wv} \\ &= \sum_w (1 - p_{uu'})p_{uw}(1 - p_{ww'})p_{wv} \\ &\geq (1 - p_{uu'}) \min_w (1 - p_{ww'}) \sum_w p_{uw}p_{wv} \\ &= (1 - p_{uu'}) \min_w (1 - p_{ww'}) p_{uv}^{(2)}, \\ \tilde{p}_{uv}^{(3)} &= \sum_x \tilde{p}_{ux}\tilde{p}_{xv}^{(2)} \\ &\geq \sum_x (1 - p_{uu'})p_{ux}(1 - p_{xx'}) \min_{\substack{w\in \\ out(x)\cap in(v)}} (1 - p_{ww'}) p_{xv}^{(2)} \\ &\geq (1 - p_{uu'}) \min_{x\in out(u)\cap in(w)} (1 - p_{xx'}) \min_w (1 - p_{ww'}) p_{uv}^{(3)} \end{aligned}$$

We observe that the minimization is taken over all walks from u to v in t steps, or more precisely, over all nodes involved in such walks, excluding u and v. Certainly, for x in such a walk, $\min_x(1 - p_{xx'}) \geq \min_{w\in V}(1 - p_{ww'})$. The other inequality can be established identically, and thus

$$\begin{aligned} (1 - p_{uu'})(\max_{w\in V}(1 - p_{ww'}))^{t-1} p_{uv}^{(t)} &\geq \tilde{p}_{uv}^{(t)} \\ &\geq (1 - p_{uu'})(\min_{w\in V}(1 - p_{ww'}))^{t-1} p_{uv}^{(t)}. \end{aligned}$$

■

Corollary 1: In particular:

1) If $p_{uu'} = a < 1$ for all $u \in V$, we have $\tilde{p}_{uv}^{(t)} = (1 - a)^t p_{uv}^{(t)}$.
2) If $p_{uv} = \frac{1}{d_{out}(u)}$ and $p_{uu'} = \frac{1}{d_{out}(u)+1}$, then $\tilde{p}_{uv} = \frac{1}{d_{out}(u)+1}$, and

$$\begin{aligned} (1 - \tfrac{1}{d_{out}(u)+1})(\max_{w\in V} \frac{d_{out}(w)}{d_{out}(w)+1})^{t-1} p_{uv}^{(t)} &\geq \tilde{p}_{uv}^{(t)} \\ &\geq (1 - \tfrac{1}{d_{out}(u)+1})(\min_{w\in V} \frac{d_{out}(w)}{d_{out}(w)+1})^{t-1} p_{uv}^{(t)}. \end{aligned}$$

Proof:

1) All inequalities in the proof of the lemma are equalities in this case.
2) By definition, $\tilde{p}_{uv} = (1 - \frac{1}{d_{out}(u)+1})\frac{1}{d_{out}(u)} = \frac{1}{d_{out}(u)+1}$.

■

The case when $p_{uu'} = a < 1$ is reminiscent of the notion of Katz centrality. A factor $(1 - a)^t$ is introduced so that the longer the path, the lower the probability. If a is chosen close to 1, e.g. $a = 0.9$, then $(1 - a)^t$ (e.g. $\frac{1}{10^t}$) becomes quickly negligible. If instead a is chosen close to 0, e.g. $a = 0.1$, then it takes longer for probabilities to become negligible (e.g. $(\frac{9}{10})^{50} \approx 0.0051$).

The case $p_{uu'} = \frac{1}{d_{out}(u)+1}$ instead uses (inverse) proportionality to the number of outgoing edges. Both the upper and lower bounds given in the proof of the above corollary depend on the degree of the nodes included in walks from u to v, and when the walk length grows, the number of distinct nodes is likely to increase, giving the bounds stated in the corollary. These bounds depend on the function $\frac{x}{x+1}$, which closely converges to 1, in fact for $d_{out}(v) = 9$, we already get 0.9. Therefore, $\tilde{p}_{uv}^{(t)}$ is mostly behaving as $p_{uv}^{(t)}$, except if $d_{out}(u)$ is small enough (say less than 8).

In summary, the emphasis of the case $p_{uu'} = a < 1$ is on the length of the walk, not on the walk itself, while for $p_{uu'} = \frac{1}{d_{out}(u)+1}$ it is actually on the nodes traversed during the walk, with the ability to separate the nodes of low entropic centrality from the others.

The bounds of Lemma 2 can be applied to the asymptotic case.

Lemma 3: The probability $\pi_{uw} = \sum_{t\geq 1} \tilde{p}_{uw}^{(t)} p_{ww'}$ to start at u and to be absorbed at $w \neq u$ over time is bounded as follows:

$$(1 - p_{uu'})(\sum_{t\geq 1} (\min_{x\in V}(1 - p_{xx'}))^{t-1} p_{uw}^{(t)}) p_{ww'} \leq \pi_{uw}$$

$$\leq \quad (1-p_{uu'})(\sum_{t\geq 1}(\max_{x\in V}(1-p_{xx'}))^{t-1}p^{(t)}_{uw})p_{ww'}.$$

Proof: Recall that $\pi_{uw} = \sum_{t\geq 1}\tilde{p}^{(t)}_{uw}p_{ww'}$ is the probability to start at u and to be absorbed at w over time. Then

$$\begin{aligned}\pi_{uw} &= \sum_{t\geq 1}\tilde{p}^{(t)}_{uw}p_{ww'} && (4)\\ &= (\tilde{p}_{uw} + \sum_{t\geq 2}\sum_{i=1}^{t-1}\tilde{p}^{(i)}_{uv}\tilde{p}^{(t-i)}_{vw} && (5)\\ &\quad + \sum_{t\geq 2}\sum_{i=1}^{t-1}\sum_{y\neq v}\tilde{p}^{(i)}_{uy}\tilde{p}^{(t-i)}_{yw})p_{ww'} && (6)\end{aligned}$$

and use Lemma 2:

$$\begin{aligned}\tilde{p}^{(i)}_{uv} &\geq (1-p_{uu'})(\min_{x\in V}(1-p_{xx'}))^{i-1}p^{(i)}_{uv},\\ \tilde{p}^{(t-i)}_{vw} &\geq (1-p_{vv'})(\min_{x\in V}(1-p_{xx'}))^{t-i-1}p^{(t-i)}_{vw}\end{aligned}$$

to get $\sum_{t\geq 2}\sum_{i=1}^{t-1}\tilde{p}^{(i)}_{uv}\tilde{p}^{(t-i)}_{vw} \geq (1-p_{uu'})(1-p_{vv'})\sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-2}\sum_{i=1}^{t-1}p^{(i)}_{uv}p^{(t-i)}_{vw}$, and in turn

$$\begin{aligned}&\sum_{t\geq 2}\sum_{i=1}^{t-1}\sum_{y\neq v}\tilde{p}^{(i)}_{uy}\tilde{p}^{(t-i)}_{yw}\\ &\quad\geq (1-p_{uu'})\sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-1}\sum_{i=1}^{t-1}\sum_{y\neq v}p^{(i)}_{uy}p^{(t-i)}_{yw}.\end{aligned}$$

Therefore

$$\begin{aligned}\pi_{uw} &\geq (1-p_{uu'})[p_{uw} + (1-p_{vv'})\\ &\quad \sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-2}\sum_{i=1}^{t-1}p^{(i)}_{uv}p^{(t-i)}_{vw}\\ &\quad + \sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-1}\sum_{i=1}^{t-1}\sum_{y\neq v}p^{(i)}_{uy}p^{(t-i)}_{yw})]p_{ww'},\end{aligned} \tag{7}$$

for v an intermediate node between u and w.

The bound can be made coarser, using that $(1-p_{vv'}) \geq \min_{x\in V}(1-p_{xx'})$:

$$\begin{aligned}\pi_{uw} &\geq (1-p_{uu'})[p_{uw} + \sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-1}\\ &\quad \sum_{i=1}^{t-1}(p^{(i)}_{uv}p^{(t-i)}_{vw} + \sum_{y\neq v}p^{(i)}_{uy}p^{(t-i)}_{yw})]p_{ww'}\\ &= (1-p_{uu'})[p_{uw} + \sum_{t\geq 2}(\min_{x\in V}(1-p_{xx'}))^{t-1}p^{(t)}_{uv}]p_{ww'}\\ &= (1-p_{uu'})\sum_{t\geq 1}(\min_{x\in V}(1-p_{xx'}))^{t-1}p^{(t)}_{uv}p_{ww'}.\end{aligned}$$

■

The bounds are well matching the intuition: three components mostly influence π_{uw}: the likelihood of leaving u (the term $1-p_{uu'}$), that of reaching w from u, and that of getting trapped at w (the term $p_{ww'}$). When $p_{uu'} = a < 1$ for all $u \in V$, the bounds on π_{uw} are met with equality and we have

$$\begin{aligned}\pi_{uw} &= (1-a)\sum_{t\geq 1}(1-a)^{(t-1)}p^{(t)}_{uw}a,\\ \pi_{uu} &= (1-a)\sum_{t\geq 1}(1-a)^{(t-1)}p^{(t)}_{uu'}a + a.\end{aligned}$$

Therefore π_{uw} is weighted by $(1-a)$ and a irrespective of the choice of u, w, and so is the corresponding entropic centrality. When a grows, π_{uu} grows, thus π_{uw} for $w \neq u$ must decrease since probabilities sum up to 1 and we have:

$$\begin{aligned}\sum_{w}\pi_{uw}\log_2(\tfrac{1}{\pi_{uw}}) &= \pi_{uu}\log_2(\tfrac{1}{\pi_{uu}}) + \sum_{w\neq u}\pi_{uw}\log_2(\tfrac{1}{\pi_{uw}}),\\ &\leq \pi_{uu}\log_2(\tfrac{1}{\pi_{uu}}) + (1-a)\log_2(\tfrac{n-1}{1-a}).\end{aligned}$$

The inequality follows from the observation that the second sum contains $n-1$ terms whose sum is at most $1-a$, and whose maximum is reached when all probabilities are $\frac{1-a}{n-1}$. Also the function $-x\log_2(x) = -\frac{1}{\ln 2}x\ln x$ is concave and has a global maximum at x such that $-\frac{1}{\ln 2}(\ln x + 1) = 0$, that is $x = \frac{1}{3}$, for which $-x\log_2(x) \approx 0.53074$. Thus

$$C^{\infty}_{H}(u) \leq 0.53074 + (1-a)\log_2(\tfrac{n-1}{1-a}). \tag{8}$$

For a given n, this upper bound becomes small when a grows. The same argument repeated for $p_{uu'} = \frac{1}{d_{out}(u)+1}$ shows (part of) the role of the out-degree of u in its centrality.

In what follows, we will provide some experiments to illustrate the consequences of choosing $p_{uu'} = a < 1$, but we will then focus on the case $p_{uu'} = \frac{1}{d_{out}(u)+1}$ for the reasons just discussed: (1) choosing $\tilde{p}_{uv}$ is influenced by the choice of the walk rather than its length, and (2) π_{uv} depends on the degrees of u and v rather than on the constant a.

To measure how central the most central node is with respect to how central all the other nodes are, Freeman introduced the notion of centralization [12].

Lemma 4: Set $n = |V|$. The asymptotic Markov entropic centralization defined by

$$C^{\infty}_{H}(G) = \frac{\sum_{v\in V}C^{\infty}_{H}(\hat{v}) - C^{\infty}_{H}(v)}{\max\sum_{v\in V}C^{\infty}_{H}(\hat{v}) - C^{\infty}_{H}(v)}$$

where $\hat{v}$ is the node with the highest Markov entropic centrality in V is given by

$$C^{\infty}_{H}(G) = \frac{\sum_{v\in V}C^{\infty}_{H}(\hat{v}) - C^{\infty}_{H}(v)}{n\log_2(n)}$$

when $p_{uu'} = \frac{1}{d_{out}(u)+1}$.

Proof: The maximum at the denominator is taken over all possible graphs with the same number of vertices. A graph that maximizes the denominator would have one node with the maximum centrality, and all the other nodes with a centrality 0 (thus minimizing the terms contributing negatively). This graph is the star graph on $|V|$ vertices, since the middle node has $|V|$ outgoing edges, and the $|V|-1$ leaves have none. Therefore all leaves have an entropic centrality of 0, and the center $\hat{v}$ of the star has maximum entropic centrality. Formally, with $|V| = n$ and putting $\hat{v}$ in the first row:

$$\hat{P} = \begin{bmatrix} \tilde{P} & D \\ 0_n & I_n \end{bmatrix},\ \tilde{P} = \begin{bmatrix} \frac{1}{n+1} & \frac{1}{n+1} & \cdots & \frac{1}{n+1} \\ 0_{n-1,1} & \frac{1}{2}I_{n-1,n-1} & & \end{bmatrix},$$

$$D = \begin{bmatrix} \frac{1}{n+1} & 0_{1,n-1} \\ 0_{n-1,1} & \frac{1}{2}I_{n-1,n-1} \end{bmatrix}.$$

Then

$$I_n - \tilde{P} = \begin{bmatrix} \frac{n}{n+1} & -\frac{1}{n+1} & \cdots & -\frac{1}{n+1} \\ 0_{n-1,1} & \frac{1}{2}I_{n-1,n-1} & & \end{bmatrix},$$

and using Schur complement:

$$(I_n - \tilde{P})^{-1} = \begin{bmatrix} \frac{n+1}{n} & \frac{2}{n} & \cdots & \frac{2}{n} \\ 0_{n-1,1} & 2I_{n-1,n-1} & & \end{bmatrix},$$

so

$$(I_n - \tilde{P})^{-1}D = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ 0_{n-1,1} & I_{n-1,n-1} & & \end{bmatrix}.$$

Lemma 1 tells us that $\pi_{\hat{v}v} = \frac{1}{n}$ for all v and thus its Markov entropic centrality is $\log_2(n)$, while $C_H^{\infty}(v)$ is 0 for $v \neq \hat{v}$. $\blacksquare$

Definition 2: Given a graph G, label its vertices in increasing order with respect to their Markov entropic centralities, that is $C_H^{\infty}(v_i) \leq C_H^{\infty}(v_{i+1})$ for $i = 1, \ldots, n-1$ and $p_{uu'} = \frac{1}{d_{out}(u)+1}$. We define *the centralization sequence* of G to be the ordered sequence

$$\left(\frac{\sum_{v\in V} C_H^{\infty}(v_i) - C_H^{\infty}(v)}{\log_2(n)}\right)_{i=1,\ldots,n}.$$

It is an increasing sequence, with values ranging from -1 to 1. We already know that the maximum is 1 by Lemma 4. The minimum is achieved when v_1 has centrality 0, and every other node has centrality $\log_2(n)/n$. This is the case if we consider a graph on n vertices defined as follows: build a complete graph on $n-1$ vertices, that is each of the $n-1$ vertices have $n-1$ outgoing edges (including to themselves). Then add one additional vertex (v_1), and an outgoing edge from every of the previous $n-1$ vertices of the complete graph to this new vertex. The advantage of studying the centralization sequence is that it captures for every node how central it is with respect to other nodes, normalized by a factor that takes into account the size of the graph.

B. A MARKOV ENTROPIC CENTRALITY ANALYSIS OF THE KARATE CLUB NETWORK

Consider the karate club network [51] (used as an example in [31]) shown in Figure 2. We use this small social network comprising 34 members (nodes) as a toy example to illustrate and validate some of the ideas explored in this paper. The 78 edges represent the interactions between members outside the club, which eventually led to a split of the club into two, and are used to predict which members will join which group. This is an undirected unweighted graph, which is treated as a particular case of directed graph ($d_{out}(u)$ is the degree of u). Let P denote the transition matrix such that $P_{uv} = \frac{1}{d_{out}(u)}$ for every $u \in V$ and every neighbor v of u. The work [31] explores the choice of D (and t) in the context of clustering. Here, we start by investigating the role of D in terms of the resulting Markov entropic centrality, for t finite ($t = 1, \ldots, 6$ since the karate club network has a diameter of 5) and asymptotic (using Lemma 1).

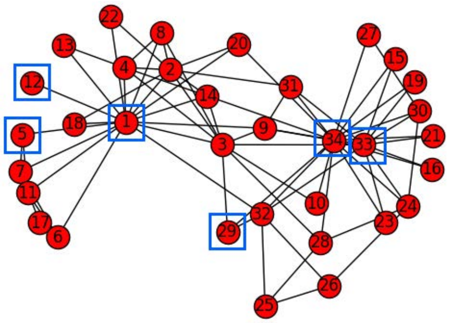


FIGURE 2. The karate club network: we show in Figure 3 the path and Markov entropic centralities (for different values of *D* and *t*) for the nodes 1, 5, 12, 29, 33 and 34. The degrees are respectively 16, 3, 1, 3, 12, 17.

a: Influence of D.

Figure 3 illustrates the Markov entropic centrality $C_H^t(u)$ of the nodes $u \in \{1, 34, 33, 29, 12, 5\}$[1] for different values of D: for $D = aI_{34}$,[2] with $a = 0.001$,[3] 0.2, 0.5, we observe that $C_H^t(u)$ is decreasing and flattening, for all nodes and for every value of t. This is expected, since, when the probabilities at the auxiliary nodes are increasing as a constant, the overall uncertainty about the random walk is reducing (as computed in (8)). Thus the higher the absorption probability, the greater the attenuation of the entropic centralities for all nodes. More precisely, (8) upper bounds $C_H^{\infty}(u)$ by

$$0.53074 + (1-a)\log_2\left(\frac{33}{1-a}\right) \approx 5.5715, 4.8238, 3.5529$$

for $a = 0.001, 0.2, 0.5$, which is consistent with the numerical values obtained ($\approx 4.8232, 4.1319, 2.9475$).

For D such that $D_{uu} = \frac{1}{d_{out}(u)+1}$ (shown on the lower right subfigure), the Markov entropic centralities are more separated than previously: indeed, a node with small degree then has a high absorption probability, which induces a large attenuation on its entropic centrality (as discussed after (8)). The net effect is a wide gulf in the centrality scores between nodes with low and high degrees.

b: INFLUENCE OF T

We notice how the centrality of a node is influenced over time by its neighbors. Consider, for example, the upper left corner figure for nodes 12 and 5. Node 12 starts (at $t = 1$) with an entropic centrality significantly lower than node 5 - indeed node 5 has three neighbors ($d_{out}(5) = 2$ neighbors and the

[1] We choose these specific nodes, since these were studied in [31].

[2] Note that I_M represents the identity matrix of dimension $M \times M$.

[3] We cannot use $D = 0$ since this means no absorption probability. Also the matrix $(I_n - \tilde{P})$ in Lemma 1 would have no reason to be invertible.

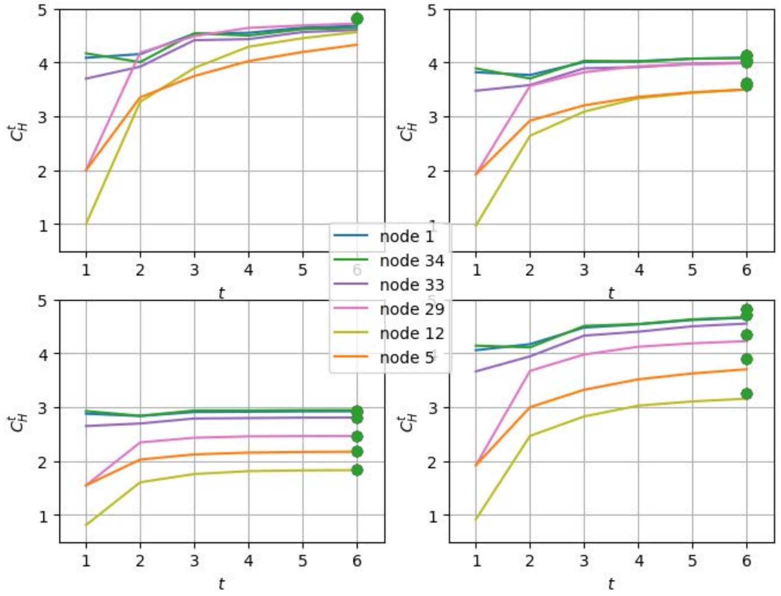


FIGURE 3. The Markov entropic centrality $C_H^t(u)$ for $u = 1, 34, 33, 29, 12, 5$ (from Figure 2) for $t = 1, 2, 3, 4, 5, 6$: on the upper left, $D = 0.001I_{34}$, on the upper right, $D = \frac{1}{5}I_{34}$, on the lower left, $D = \frac{1}{2}I_{34}$ and for D such that $D_{uu} = \frac{1}{d_{out}(u)+1}$ on the lower right. The dots show the asymptotic values $C_H^\infty(u)$.

self-loop), more than 12 which has only one (thus $C_H^1(12) = 1(1/2)\log_2(2)$ including the self-loop). Yet node 12 reaches a ''bridge'' (a node of high centrality), namely node 1, at $t = 2$, and thus for $t \geq 2$, its entropic centrality grows, and eventually ends up being almost as high as that of node 1 itself. In contrast, even though node 5 reaches the same bridge, it also belongs to a local community within the graph, inside which a significant amount of its influence (flow) stays confined. This explains why node 5 ends up having a lower entropic centrality than node 12 in particular. In the upper right and lower left plots, we have assigned a significant volume of the flow to be absorbed at the auxiliary nodes, which has a net effect of attenuating the absolute values of entropic centrality for all nodes, i.e., a downward shift and flattening. This happens for nodes 5, 12, 29 on the lower right plot, since $\frac{1}{d_{out}(u)+1}$ is significant if the node has a low degree. Taken together with the initial gap among the centralities of nodes 5 and 12, we thus do not observe the overtake in these experiments, unlike in the case where the absorption probability was negligible. We finally observe that C_H^t may vary for small values of t but becomes consistent when t grows.

c: CENTRALIZATION SEQUENCE

The min and the max of the centralities are 3.0859 and 4.7216, while the mean and the median are 4.07636 and 3.9111. The min, median, mean and max of the centralization sequence as defined in Definition 2 are [-0.19467,-0.03248,0,0.12682]. The values for the above studied nodes of interest are 0.1226 for 1, 0.1241 for 34, 0.10195 for 33, 0.03485 for 29, -0.17471 for 12 and -0.06289 for 5. In this list, nodes 1 and 34 are thus considered as most central (which can be seen from Figure 2) , however centralization further measures the extent to which they are most central, e.g. respectively 0.1226 and 0.1241, i.e., ~ 0.12 from the mean.

The above analysis suggests that changes in the Markov entropic centralities over time are indicative of local communities in the graph, with changes in gradient corresponding to traversal of boundaries from one community to another. Nodes with low centralization have a low centrality with respect to both other nodes in the graph and graphs of the same size, since they are likely to be either isolated or to belong to a small community (e.g. node 5). While the reverse argument suggests that nodes with high centralization (e.g. nodes 1, 33 and 34) are likely to be either bridges or close to bridges. We will explore and exploit this observation to design a clustering algorithm in Section IV, where the community structures are first explored around low centrality nodes.

d: COMPARISON WITH KNOWN CENTRALITIES

Table 1 compares different centralities. In addition to the previously mentioned prominent centrality measures, we also consider load centrality [16] in our experiments, because it captures the fraction of traffic that flows through individual nodes (load), considering all pairwise communications to be through corresponding shortest paths. Both the path and the asymptotic Markov centralities give the same ranking, it is similar to the ranking given by the degree centrality C_D (with

TABLE 1. Different centrality measures: C_H and C_H^∞ are the path [48] and asymptotic Markov entropic centralities (1), C_D, C_B, C_L are respectively the degree, betweenness and load centralities.

node	C_H	C_H^∞	C_D	C_B	C_L
node 34	4.83992	4.82504	0.5151	0.3040	0.2984
node 1	4.83041	4.81999	0.4848	0.4376	0.4346
node 33	4.76892	4.72539	0.3636	0.1452	0.1476
node 29	4.50092	4.34323	0.0909	0.0017	0.0017
node 5	4.07244	3.90674	0.0909	0.0006	0.0006
node 12	3.39469	3.26763	0.0303	0	0

some difference regarding node 5 and 12, which are explained by the above discussion). The betweenness and load centrality agree on their ranking, which is different from the other ones. These results are not surprising: for a small graph, the degree heavily influences short paths/walks, explaining the agreement among C_H, C_H^∞ and C_D, but it has little to do with the betweenness and the load. The same ranking observed between betweenness and load centrality is expected, since the latter was proposed to be a different interpretation of betweenness centrality [16], even though some minor differences have been identified subsequently [7].

III. ENTROPIC CENTRALITY FOR WEIGHTED GRAPHS

Consider a weighted directed graph $G_w = (V, E)$, where the weight function $w : E \to \mathbb{R}_{\geq 0}$ attaches a non-negative weight $w(u, v)$ to every edge $(u, v) \in E$. For a node $u \in V$, let $\mathcal{N}_u = \{v \in V,\ (u, v) \in E\}$ be the set of out-neighbors of u, $d_{out}(u) = |\mathcal{N}_u|$ be the out-degree of u, and $d_{w,out}(u) = \sum_{v\in\mathcal{N}_u} w(u, v)$ be the weighted out-degree of u.

A natural way to define a transition matrix P_w to describe a random walk over G_w taking into account the weight function w would be to set $P_{w,uv} = \frac{w(u,w)}{d_{w,out}(u)}$. It is however known that adapting centrality measures for unweighted graphs to weighted graphs in that manner comes at the risk of changing their meaning, see e.g. [36],and that one way to remedy this is by the introduction of a weight parameter.

A. WEIGHTED MARKOV ENTROPIC CENTRALITY

In order to capture the weights of the outgoing edges in the current framework of Markov entropic centrality, we introduce two tuning parameters, a conversion function $\alpha : w(e) \to \alpha(w(e))$ and a node weight function $\mu : v \to \mu(v)$.

The conversion function $\alpha : w(e) \to \alpha(w(e))$ adjusts the transition matrix $P_{\alpha(w)}$ such that $P_{\alpha(w),uv} = \frac{\alpha(w(u,w))}{d_{\alpha(w),out}(u)}$ (with $d_{\alpha(w),out}(u) = \sum_{v\in\mathcal{N}_u} \alpha(w(u, v)))$ depending on the importance that weights are supposed to play compared to edges. The two obvious choices for α are $\alpha(w(e)) = 1$ for all edges e (reducing to the unweighted case), and $\alpha(w(e)) = w(e)^\beta$ for some parameter β. This formulation has the flexibility to give more or less importance to weights with respect to edges.

To address the issue of defining entropic centrality for weighted graphs, we need a tuning parameter (e.g., [36]) within the definition of entropic centrality, that maintains the semantics and meaning of the notion of entropy. We use the node weight function $\mu : v \to \mu(v)$ and propose the notion of weighted Markov entropic centrality, which is inspired by the concept of weighted entropy [17].

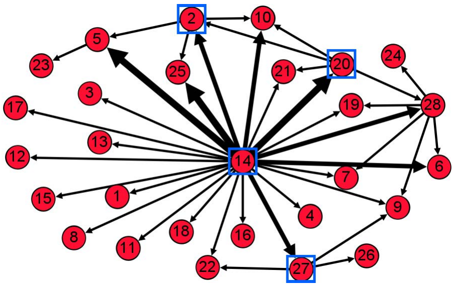


FIGURE 4. The cocaine dealing network [11]: weighted edges are drawn with quantized girth. Figure 5 shows the (weighted) Markov entropic centrality of the nodes 2, 27, 20 and 14.

Definition 3: The *weighted Markov entropic centrality* $C^t_{\alpha(w),H}(u)$ of a node u at time t is defined to be

$$-\sum_{v\in V}\mu(v)(\tilde{p}^{(t)}_{\alpha(w),uv}+p^{(t)}_{uv'})\log_2(\tilde{p}^{(t)}_{\alpha(w),uv}+p^{(t)}_{uv'}), \qquad (9)$$

where $\tilde{p}^{(t)}_{\alpha(w),uv}$ is the probability to reach v at time t from u, for v in V, taking into account the weights $\alpha(w(e))$ for every edge in E. Auxiliary nodes defined for the unweighted case are still present and $p^{(t)}_{uv'}$ is the probability to reach an auxiliary node v' from u at time t, which depends on the absorption probability matrix D, as in the unweighted case. The probabilities $\tilde{p}^{(t)}_{\alpha(w),uv}+p^{(t)}_{uv'}$ are obtained from the matrix $\hat{P}^t_{\alpha(w)}$ (using $P_{\alpha(w)}$ instead of P in Lemma 1).

Before discussing the choice of $\mu(v)$, we recall that when $t \to \infty$, as for the unweighted case, the terms in column v of $\hat{P}^t_{\alpha(w)}$ become 0, and we are left with the terms in column v', of the form

$$\pi_{\alpha(w),uv} := \begin{cases} \sum_{t\geq 1}\tilde{p}^{(t)}_{\alpha(w),uv}p_{vv'} & u \neq v \\ (\sum_{t\geq 1}\tilde{p}^{(t)}_{\alpha(w),uv}+1)p_{vv'} & u = v, \end{cases}$$

asymptotically yielding the entropic centrality

$$C^\infty_{\alpha(w),H}(u) = -\sum_{v\in V}\mu(v)\pi_{\alpha(w),uv}\log_2(\pi_{\alpha(w),uv}).$$

Lemma 3 holds, therefore, similarly to the unweighted case, we have that if the probability of absorption is $p_{uu'} = a$, then repeating the arguments leading to (8) yields:

$$\begin{aligned}\sum_w \mu(w)\pi_{uw}\log_2(\tfrac{1}{\pi_{uw}}) & \qquad (10)\\ = \mu(u)\pi_{uu}\log_2(\tfrac{1}{\pi_{uu}}) + \sum_{w\neq u}\mu(w)\pi_{uw}\log_2(\tfrac{1}{\pi_{uw}}), & \\ \leq \max_{v\in V}\mu(v)(0.53074 + (1-a)\log_2(\tfrac{n-1}{1-a})). & \qquad (11)\end{aligned}$$

This shows that while $\max_{v\in V}\mu(v)$ may increase the overall centrality, it remains true, as for the unweighted case, that increasing a just reduces the overall centrality. Therefore we will continue to use $p_{uu'} = \frac{1}{d_{out}(u)+1}$ as absorption probability.

The computation of $C^t_{\alpha(w),H}(u)$ involves a sum over all nodes $v \in V$, including $v = u$, and the probability to go

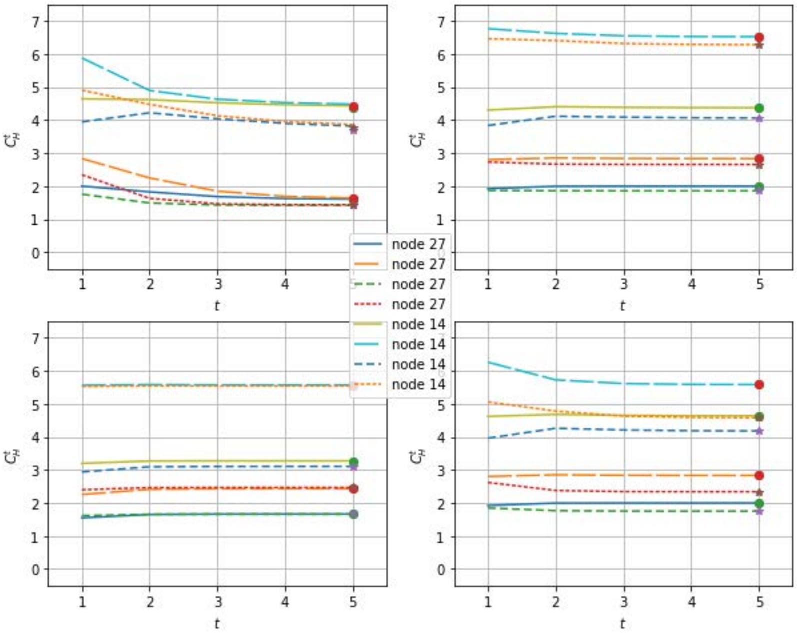


FIGURE 5. **The Markov entropic centrality $C^t_{\alpha(w),H}(u)$ for $u = 27, 14$ (from Figure 4) for $t = 1, \ldots, 5$: on the upper left, $D = 0.001I_{28}$, on the upper right, $D = 0.2I_{28}$, on the lower left, $D = 0.5I$ and D with $D_{uu} = \frac{1}{d_{w,out}(u)+1}$ on the lower right.**

from u to every v. The weight $\mu(v)$ captures the influence of the reached node v: if we reach influential nodes v of weight $\mu(v)$ in t time, then u should be more central than if the only nodes we reach from it have no influence themselves. We define $\mu(v)$ to be $\mu(v) = \left(\frac{d_{w,out}(v)}{d_{out}(v)}\right)^{\gamma}$ if $d_{out}(v) \neq 0$. When $d_{out}(v) = 0$, $d_{w,out}(v) = 0$ and we set $\mu(v) = 1$. The weighted Markov entropic centrality (9) gives an influence which is proportional to the ratio $\left(\frac{d_{w,out}(v)}{d_{out}(v)}\right)^{\gamma}$, while γ gives a way to amplify or reduce this influence.

If all weights are 1, then the ratio simplifies to 1, and we are back to the non-weighted definition of Markov entropic centrality (as also with $\gamma = 0$). Other possible choices for $\mu(v)$ could be explored, such as $\mu(v) = \frac{\log_2 d_{w,out}(v)}{\log_2 d_{out}(v)}$. Indeed, $\log_2 d_{w,out}(v)$ would be the influence that v would have had over its neighbors in terms of entropic centrality if it had $d_{w,out}(v)$ neighbors of weight 1, while $\log_2 d_{out}(v)$ is the entropic centrality over the neighbors of v, ignoring the weights.

If the weights are normalized such that the lowest weight is mapped to 1, the entropic centrality values obtained when using a non-trivial function for $\mu(v)$ will necessarily be at least as much as obtained with $\mu(v) = 1$.

The centralization as computed in Lemma 4 can be generalized to the weighted case. In the unweighted case, comparison was done among graphs with the same number of vertices. In the weighted case, comparison is done among graphs with the same number of vertices, with given weights $\mu(v)$. A graph that would give the maximum centrality consists again of a star graph, with center v_1, assuming that the leaves have a self-loop which can be weighted, in which case $d_{out}(v_i) = 1$ for $i \neq 1$, and $\mu(v_i)$ is any weight. The centrality of the leaves will be zero, but that of v_1 will be $\sum_{i=1}^{n} \mu(v_i) p_i \log_2(1/p_i)$. However maximizing this quantity over p_i does not have a closed form expression. Loose bounds can be computed though: $\frac{1}{n}\log(n)\sum_{i=1}^{n}\mu(v_i) \leq \max_{p_1,\ldots,p_n}\sum_{i=1}^{n}\mu(v_i)p_i\log_2(1/p_i) \leq 0.53074\sum_{i=1}^{n}\mu(v_i)$ (see before (8) for a bound on $p\log_2(1/p)$).

B. A WEIGHTED MARKOV ENTROPIC CENTRALITY ANALYSIS OF A COCAINE DEALING NETWORK

We consider the cocaine dealing network [11], a small directed weighted graph obtained from an investigation into a large cocaine traffic in New York City. It involves 28 persons (nodes), and 40 directed weighted edges representing communication exchanges obtained from police wiretapping.

The (weighted) Markov entropic centrality depends on the choice of the absorption matrix D. We thus start by looking at how a change in D influences the entropic centralities. We keep the same choices for D as for the unweighted case, namely $D = 0.001I_{28}$, $D = 0.2I_{28}$, $D = 0.5I$ and D such that $D_{uu} = \frac{1}{d_{w,out}(u)+1}$,since, if $w(e) = 1$ for all edges, then $d_{w,out}(u) = d_{out}(u)$ for all nodes.

In Figure 5, we plot the (weighted) Markov entropic centrality $C^t_{\alpha(w),H}(u)$ for $u = 27, 14$ for $t = 1, 2, 3, 4, 5, 6$, for the 4 choices of absorption probabilities D. For $u = 27, 14$, four variations of Markov entropic centralities are considered: $\alpha(w) = 1$ and $\mu(v) = 1$ (straight lines), corresponding to the unweighted case, $\alpha(w) = 1$ and $\mu(v) = \left(\frac{d_{w,out}(v)}{d_{out}(v)}\right)$

(long dash lines) for the case where the transition matrix P is used, $\alpha(w) = w$ and $\mu(v) = 1$ (short dash lines) to show how centrality is computed based purely on P_w, and finally $\alpha(w) = w$ with $\mu(v) = \left(\frac{d_{w,out}(v)}{d_{out}(v)}\right)$ (dotted lines), for which P_w is used for the transition matrix, together with the weighted entropic centrality. Nodes 27 and 14 are different in nature, in that node 14 acts as a bridge, with high degree compared to the rest of the other nodes, while node 27 has only one incoming edge and two outgoing edges. They are chosen as representatives of nodes of respectively high and low degree (even though the degree is not the only contributing factor in the node centrality, it still plays an important role, particularly for small values of t and more generally in small graphs). For each of the 4 plots, the 4 upper lines characterize the behavior of node 14, and the 4 lower lines, that of node 27. We observe that the behavior of the entropic centralities when $\mu(v) = 1$ are similar to what was already observed for the karate club network: for $D = 0.2I_{28}$ and $D = 0.5I$, the centralities are flattened because the path uncertainty is reduced when the absorption probabilities are increasing (as shown in (11)), while for D such that $D_{uu} = \frac{1}{d_{w,out}(u)+1}$, the gap among the centralities is (slightly) increasing. When $\mu(v) = \frac{d_{w,out}(v)}{d_{out}(v)}$, both centralities are amplified. Since node 14 has many outgoing edges, with weights including 10, 11, 14 18, 19, the introduction of $\mu(v)$ creates a higher amplification for node 14 than for node 27, whose edge weights are all less than 5. Zooming at time $t = 2$ for node 14 and $D = 0.001I_{28}$, we notice that the short dashed line has a peak, before going down. This is explained by the fact that when $\alpha(w) = w$ and $\mu(v) = 1$, the entropic centrality is that of a node with degree amplified by the weights, thus creating an initial jump in the entropy. However at the next time, several reached nodes have in turn very few (or no) neighbors, and edges of low weights, and this leads to a saturation effect. This behavior is less prominent for other choices of D, where the absorption probability is high, and hence the effect of the edge weights is attenuated.

We then focus on the case where $D_{uu} = \frac{1}{d_{w,out}(u)+1}$, which depends only on the network setting, instead of other choices of D which are parameters whose range can be explored one by one. With this choice of D, we consider the (weighted) Markov entropic centrality for $u = 2, 27, 20$, as displayed in Figure 6. These nodes have outgoing edges with weights 1, 1, 1, for $u = 2$, weights 4, 3, 1 for $u = 27$, and weights 2, 1, 1, 1 for $u = 20$. These nodes are chosen because they have similar out-degrees, but different weighted out-degrees. The same 4 scenarios as above are considered. For node 2, its edge to 10 stops at 10, its edge to 5 has only one connection to node 23 which has weight 1, and its last edge goes to 25, which has no connection either, therefore the 4 centralities are actually of the same quantity, and thus all the 4 lines are coincident. For node 27, since it has the same out-degree as node 2, in the unweighted case, it starts at the same centrality as node 2. However it is even less influential since none of its own neighbors have neighbors. In the 3rd scenario, its entropic centrality is even lower than in the unweighted case, which is normal, since in this case, the walk distribution is not uniform anymore. In the two cases where $\mu(v) = \frac{d_{w,out}(v)}{d_{out}(v)}$, we see a jump in the centrality, explained by the weights of the outgoing edges which are higher than for node 2. However, the centrality of node 27 remains below that of 20, whose weighted out-degree is less than that of 27, but its out-degree is actually more.

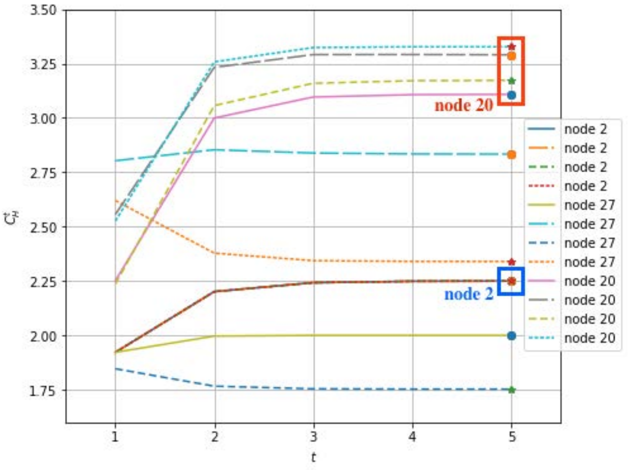


FIGURE 6. The Markov entropic centrality $C^t_{\alpha(w),H}(u)$ with $D_{uu} = \frac{1}{d_{w,out}(u)+1}$: for $u = 2, 27, 20$ (from Figure 4), for $t = 1, \ldots, 5$ and for $(\alpha(w), \mu(v)) = (1, 1)$ (straight lines), $(1, \frac{d_{w,out}(u)}{d_{out}(u)})$ (long dash lines), $(w, 1)$ (short dash lines) and $(w, \frac{d_{w,out}(u)}{d_{out}(u)})$ (dotted lines).

C. INTERPRETING ENTROPIC CENTRALITY WITH A BITCOIN SUBGRAPH

We next consider a small subgraph extracted from the bitcoin network comprising 178 nodes and 250 edges. Nodes are bitcoin addresses, and there is an edge between one node to another if a bitcoin payment has been made. Since the proposed Markov entropic centrality measure is suitable for undirected, directed and weighted graphs, we look at the chosen bitcoin subgraph in different variations: (i) as an undirected unweighted graph by ignoring the direction of transactions, (ii) as a directed unweighted graph by just considering whether any transactions exists, and (iii) as a directed weighted graph (with $\alpha(w(e)) = w(e)$, $\mu(v) = \frac{d_{w,out}(v)}{d_{out}(v)}$) to capture the effect of transaction amount.

In Figure 7 we show the three variations with nodes colored according to their Markov entropic centrality at $t = \infty$ (asymptotic). The darker the node color, the higher the Markov entropic centrality. In subfigure 7a, one node stands apart, with the highest entropic centrality, which is a node which is highly connected to other nodes. If we look at the same node in subfigure 7b, we see that it is not central anymore: this is because, in the directed graph, this node is actually seen to receive bitcoin amounts from many other nodes, so, as far as sending money is concerned, it has

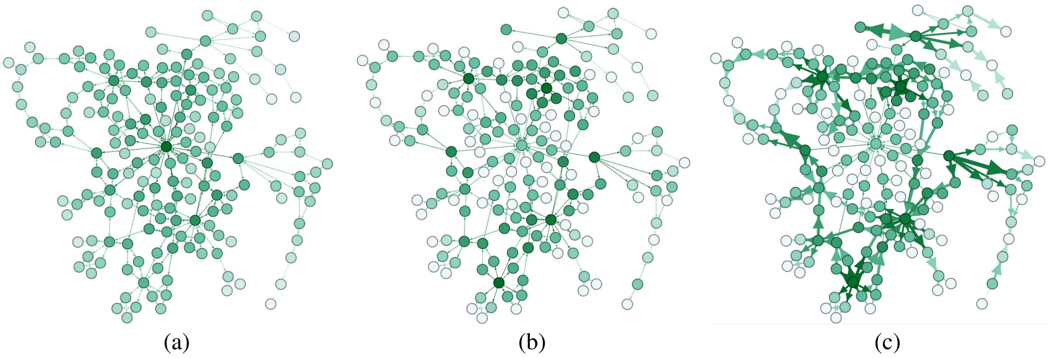


FIGURE 7. Asymptotic Markov entropic centralities for a 178 node subgraph of the bitcoin network: undirected and unweighted on the left, directed and unweighted in the middle, directed and weighted on the right (the highest weight edges are emphasized). The top 10%, next 20%, 30% and remaining 40% nodes in terms of their entropic centrality are shown in progressively lighter colors.

very little actual influence (however, in the same graph, but with edge directions reversed, this node would have had the highest centrality). On the other hand, nodes that were not so prominent in the undirected graph appear now as important. The graph in subfigure 7b highlights nodes which are effectuating many bitcoin transactions. Now one node may do several transactions with little overall bitcoin amount, but we may want to identify a node that does few, but high amount transactions. This is illustrated in subfigure 7c. It turns out that for the subgraph we studied, high centrality nodes in the unweighted case remain high centrality nodes in the weighted case, indicating that nodes performing many bitcoin transactions happen also to be those carrying out high overall amount of transactions.

For the sake of completeness, we provide Kendall-τ rank correlation coefficients among different variations of (weighted) Markov entropic centralities for the directed graph, for $t = \infty$: for $(\alpha(w(e)) = 1, \mu(v) = 1)$ (shown on subfigure 7b), for $(\alpha(w(e)) = w(e), \mu(v) = \frac{d_{w,out}(v)}{d_{out}(v)})$ (shown on subfigure 7c), but also for $(\alpha(w(e)) = w(e), \mu(v) = 1)$, and for $(\alpha(w(e)) = 1, \mu(v) = \frac{d_{w,out}(v)}{d_{out}(v)})$. The out-degree centrality is also tested. A Kendall-τ coefficient is a measure of rank correlation: the higher the measure, the higher the pairwise mismatch of ranks across two lists. Results are shown in Table 2. Above the diagonal, coefficients are obtained using all 178 nodes. Below the diagonal, only 67 nodes are used, which are in the union of the top 20% nodes for each of the 5 aforementioned centrality measures. In the given graph, many (peripheral) nodes have the same (low) entropic centrality, so when all nodes are considered for ranking correlation, the average is misleadingly low. Since high centrality nodes are often of interest, ranking variations among the top nodes is pertinent, and we observe that (i) out-degree is not a good proxy for most entropic centrality variants, and (ii) the different variations yield significantly distinct sets of high centrality nodes, corroborating the need for our flexible entropic centrality framework.

TABLE 2. Kendall-τ coefficients of 5 centrality measures: 4 of them are the Markov entropic centralities for different values of $\alpha(w(e))$ and $\mu(v)$; d_{out} is the out-degree centrality. Above the diagonal, coefficients are obtained using all 178 nodes. Below the diagonal, only 67 nodes are used, which are in the union of the top 20% nodes for each of these five centrality measures.

α,μ	1,1	d_{out}	w,1	$1, \frac{d_{w,out}(v)}{d_{out}(v)}$	$w, \frac{d_{w,out}(v)}{d_{out}(v)}$
1,1	0	0.159	0.075	0.084	0.086
d_{out}	0.419	0	0.193	0.214	0.218
w,1	0.315	0.60	0	0.120	0.050
$1, \frac{d_{w,out}(v)}{d_{out}(v)}$	0.186	0.538	0.305	0	0.089
$w, \frac{d_{w,out}(v)}{d_{out}(v)}$	0.346	0.671	0.097	0.283	0

IV. ENTROPIC CENTRALITY BASED CLUSTERING

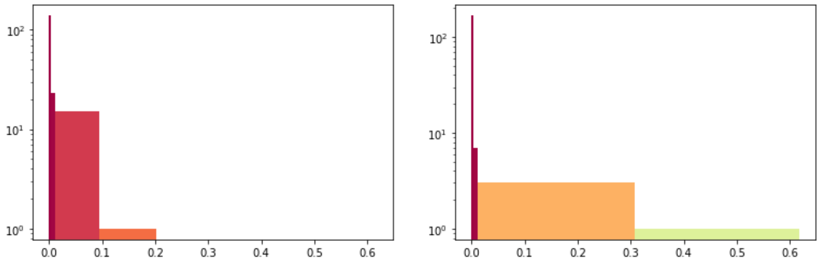
We next explore whether and how the local communities inferred from the gradients observed in the evolution of the entropic centralities as a function of t (see Section II) can be exploited to realize effective clustering.

The formulas (1) and (9) for the (weighted) Markov entropic centralities involve the sum of probabilities $p_{uv}^{(t)}+p_{uv'}^{(t)}$ and $p_{\alpha(w),uv}^{(t)} + p_{\alpha(w),uv'}^{(t)}$ respectively. We will use $\hat{P}$ to denote the matrix whose uth row contains, in column v, the coefficient $p_{uv}^{(t)}+p_{uv'}^{(t)}$ in the unweighted case, and $p_{\alpha(w),uv}^{(t)}+p_{\alpha(w),uv'}^{(t)}$ in the weighted case. Since the algorithm that we describe next uses $\hat{P}$ in the same manner, irrespective of t (though we will focus on the case $t = \infty$) or whether there is a weight function, we keep the same notation $\hat{P}$.

The proposed algorithm works in two stages: first, we create local clusters around 'query nodes', and then, we carry out a hierarchical agglomeration. The choice of query nodes in our approach is informed by the entropic centrality, which we describe first (Subsection IV-A), before we elaborate the algorithm (in Section IV-B).

A. QUERY NODE SELECTION

In the initial step of the algorithm, we look for a cluster around and inclusive of a specific query node, similarly to [4]. In [4] though, it was not possible a priori to determine suitable query nodes (e.g., a node could be at the boundary of two



(a) A high centrality query node.

(b) A low centrality query node.

FIGURE 8. **Histogram showing number of nodes (*y*-axis in log-scale) at which a random walker is absorbed for given (range of) values of absorption probability (*x*-axis) for the 178 node Bitcoin subgraph, treated as unweighted and undirected.**

clusters, and thus yield the union of subsets of these clusters as a resultant cluster), and hence multiple constrained random walkers were deployed to increase the confinement probability of the random walks within a single cluster. In contrast, our choice of query nodes is informed by their Markov entropic centrality.

Let us first formalize some of the information captured by the notion of entropy.

Proposition 1: Given the probabilities $p_1 \leq p_2 \leq \ldots \leq p_m < p_{m+1} = \frac{1}{n} = \ldots = p_{m+s} < p_{m+s+1} \leq \ldots \leq p_n$, $p_i \geq 0$, $\sum_{i=1}^{n} p_i = 1$, and $\gamma > 1$ such that $\sum_{i=1}^{n} p_i \log_2(1/p_i) = \frac{\log_2(n)}{\gamma}$, we have:

1) $s \leq \frac{n}{\gamma}$, and $n - m - s \geq n\frac{\tau-1}{\tau}\frac{\gamma-1}{\gamma}$ for $\tau > 1$,
2) $\sum_{i=m+s+1}^{n} p_i > \frac{\tau-1}{\tau}\frac{(\gamma-1)}{\gamma} \iff \sum_{i=1}^{m} p_i < 1 - \frac{s}{n} - \frac{\tau-1}{\tau}\frac{(\gamma-1)}{\gamma}$.

Proof: Since we ordered the n probabilities by increasing order, we write $\sum_{i=1}^{n} p_i \log_2(1/p_i) = \sum_{i=1}^{m} p_i \log_2(1/p_i) + \frac{s}{n}\log_2(n) + \sum_{i=m+s+1}^{n} p_i \log_2(1/p_i)$, and $\sum_{i=1}^{n} p_i \log_2(1/p_i) = \frac{\log_2(n)}{\gamma}$ for some $\gamma > 1$ certainly implies $\frac{s}{n}\log_2(n) \leq \frac{\log_2(n)}{\gamma}$. Thus there must be $s \leq \frac{n}{\gamma}$ probabilities of the form $\frac{1}{n}$ (in particular, if $\gamma > n$, $s = 0$, and likewise, when $\gamma > 1$, $s \leq n-1$). In turn, we must have $n-s$ which is at least $\frac{n(\gamma-1)}{\gamma}$ probabilities shared between $p_1, \ldots, p_m$ and $p_{m+s+1}, \ldots, p_n$, and they must sum up to $1 - \frac{s}{n} = \frac{n-s}{n}$.

It is not possible that all $n-s$ probabilities belong to the group $p_1, \ldots, p_m$ (i.e., $< \frac{1}{n}$), because if all of them were strictly less than $\frac{1}{n}$, we would have at most $\frac{s}{n} + (n-s)p_m < \frac{s}{n} + \frac{n-s}{n} < 1$ (a similar argument holds for not having all the probabilities in the group $p_{m+s+1}, \ldots, p_n$). More generally, this is saying that deficiency from $1/n$ among the $p_1, \ldots, p_m$ has to be compensated by the $p_{m+s+1}, \ldots, p_n$:

$$\sum_{i=1}^{m} p_i = 1 - \frac{s}{n} - \sum_{i=m+s+1}^{n} p_i,$$

for $0 < \sum_{i=1}^{m} p_i < \frac{n-s}{n}$, and accordingly $\sum_{i=m+s+1}^{n} p_i$ varies from being strictly less than $1 - \frac{s}{n}$ to strictly more than 0.

We can refine these ranges using the number of probabilities we are allowed in each sum. Suppose that among the $n-s$ probabilities that are left to be assigned, $\frac{1}{\tau}(n-s)$ are the probabilities $p_1, \ldots, p_m$ for some $\tau > 1$, and the rest, namely $\frac{\tau-1}{\tau}(n-s) \geq \frac{\tau-1}{\tau}\frac{n(\gamma-1)}{\gamma}$ are the probabilities $p_{m+s} < p_{m+s+1} \leq \ldots \leq p_n$.

For at least $\frac{\tau-1}{\tau}\frac{n(\gamma-1)}{\gamma}$ probabilities strictly more than $1/n$, we have:

$$\sum_{i=m+s+1}^{n} p_i > \frac{\tau-1}{\tau}\frac{(\gamma-1)}{\gamma} \iff \sum_{i=1}^{m} p_i < 1 - \frac{s}{n} - \frac{\tau-1}{\tau}\frac{(\gamma-1)}{\gamma}.$$

■

We apply this proposition to π_{uv}, for a given node u, and for v ranging over possible nodes in V. Let $\tau > 1$ be a parameter that decides the proportion of probabilities less than $\frac{1}{|V|}$ (the larger τ, the smaller the proportion), and let $\gamma > 1$ be a parameter that characterizes how small the Markov entropic centrality of a node u is (the larger γ, the lower the entropic centrality). The result says that given γ, there is a tension between agglomeration of small probabilities on the one hand and agglomeration of large probabilities on the other hand: the larger the agglomeration of large probabilities (as a function of τ and γ), the lower the agglomeration of small probabilities, and vice-versa. Furthermore, as γ grows, it creates an agglomeration of large probabilities which is detached from that with small probabilities. This suggests that nodes with a low entropic centrality have local clusters where ties are relatively strong, while nodes tend to have a relatively high entropic centrality when they either are embedded within a large cluster, or when they are at the boundary of two clusters (and could arguably belong to either). Accordingly, we choose to start identifying local communities by choosing the low entropic centrality nodes as query nodes.

This is illustrated in Figure 8 for the 178 node Bitcoin subgraph, treated as unweighted and undirected. We consider a node with high centrality on the left hand-side, it has centrality $\approx$ 5.08297, and one with low centrality on the right hand-side (with centrality $\approx$ 1.73539). By high and low, we mean that both belong to the top 10 nodes in terms of respective high/low centrality. The histograms show the probability that a random walker is (asymptotically) absorbed at a node. The y-axis is in logscale. The bins are placed at 0, $1/2|V|$, $2/|V|$, and then at half the maximum probability (as observed across the two cases) and the maximum probability.

In the first bin, for both cases, there are more than 100 nodes which have a negligible probability of the random walker being absorbed. As predicted by Proposition 1, for the node with high entropic centrality, they are less spread apart (and there are more probabilities close to $1/|V|$) than for the low entropic centrality node, which has, furthermore one high probability (between 0.3 and 0.6). Such higher probabilities of absorption at a few nodes in the graph are precisely what we use as a signal for a local community structure.

This finding is consistent with the observation at the end of Section II-B, that if there is no significant change in the entropic centrality over time, then the node belongs to a small community, while larger entropic centralities indicate nodes that are well connected (possibly indirectly) to many nodes rather than being strongly embedded in a particular well defined community. In Figure 9 we show a scatter diagram illustrating, again for the 178 Bitcoin subgraph, the relations between the entropic centrality of a node and the highest probability of absorption at any node for a random walker starting at the respective node. We see that nodes with high centralities have as highest absorption probability, values below 0.3, while low entropic centrality nodes have highest absorption probability of more than 0.4. For this subgraph, the centralization sequence has a minimum of -0.27962, a maximum of 0.40350, a mean of 0 and a median of 0.00560. The nodes that are "circled" in red have highest centralization (centralization values more than 0.18), and those with lowest centralization (centralization values less than -0.2) are encircled in blue.

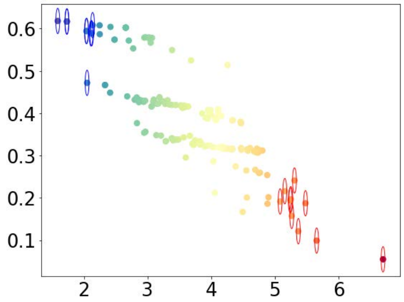


FIGURE 9. On the x-axis, the centrality of nodes, on the y-axis, the value of the highest probability of absorption at any node, for a random walker starting at that node (in the 178 node Bitcoin subgraph, treated as unweighted and undirected). The two vertical bars demarcate the top 30% and 40% of nodes in terms of entropic centrality. Nodes that are circled have highest/lowest centralization.

The Markov entropic centrality score acts as a good summary of the detailed probability distributions of absorption for a random walker starting at different query nodes. Specifically, we see that using low entropic centrality nodes as query nodes would yield meaningful local clusters precisely by identifying the nodes at which the random walker is absorbed with high probability to constitute a local community. Such communities can then be further coalesced into larger (and fewer) clusters or re-clustered into smaller (and more numerous) communities, as deemed suitable, in a hierarchical manner. We will see, in our experiments, that, what is a suitable 'low' or 'high' entropic centrality, depends on the distribution of the centrality scores. The distribution of the relative Markov centrality scores for a given graph is thus also a good indicator of whether the proposed approach would be effective for clustering that graph instance, and helps us reason about whether to use the proposed approach. This is illustrated with experiments, in particular in Subsection IV-F.

B. ENTROPIC CENTRALITY BASED CLUSTERING ALGORITHMS

Keeping in mind the above discussion, we describe the ***first stage*** (without the iterative hierarchical process) of clustering in Algorithms 1 and 2.

Algorithm 1 Probability Distribution Based Graph Clustering

```
procedure ProbDistClustering($G = (V, E), N, t$)
    ▷ $N \ll |V|$ stands for the top-$N$ most central nodes
    Compute $\hat{P}$ and the entropic centrality of all $v \in V$.
    Assign $S_{HE}$ = {the top-$N$ entropic centrality nodes}.
    ▷ Initialization
    Set $Q$: nodes in ascending order of entropic centrality.
    Set $S_{clust} = \emptyset$.                    ▷ Current clusters
    ▷ Global clustering
    while $Q$ is not empty do
        Take the query node $v_q$ from $Q$'s head (remove it).
        ▷ Obtain query node centric local cluster
        Apply a(ny) clustering algorithm on the row $(\hat{P}_{v_q,v})_{v \in V}$ of $\hat{P}$ to form the set $S_{ini,v_q}$ of clusters.
        Choose $S_{v_q}$ from $S_{ini,v_q}$ with the highest average transition probability (include $v_q$).    ▷ $v_q$'s raw cluster
        ▷ Prune the raw cluster $S_{v_q}$ using Algorithm 2.
        ProcessRawCluster($S_{v_q}, S_{HE}, S_{clust}$)
        $\forall v \in S_{v_q}$, remove $v$ from $Q$.
        ▷ Integrate the local result with the global view.
        if $S_{v_q}$ intersects with any cluster(s) in $S_{clust}$ then
            Merge them (update $S_{clust}$ accordingly).
        else Add $S_{v_q}$ to $S_{clust}$.
    return $S_{clust}$
```

In Algorithm 1, we maintain S_{clust} as a current global view of clusters. We start from the lowest entropic centrality node as a query node, and repeat the process as long as there are nodes that do not already belong to some existing cluster (listed in S_{clust}).

We consider the transition probabilities from a query node v_q to all the other nodes as per $\hat{P}$, and carry out a clustering of these (one-dimensional, scalar) probability values. Lemma 1 shows the existence of (up to) "three clusters", formed by probabilities of values around $1/|V|$, and of values away from $1/|V|$, either by being smaller or larger enough. How clearly defined these clusters are depends on γ: if γ is small, close to 1, probability values can still be close to uniform, on the

Algorithm 2 Pruning of the Raw Cluster S_{v_q}

```
procedure ProcessRawCluster($S_{v_q}$, $S_{HE}$, $S_{clust}$)
▷ most central node list $S_{HE}$, current cluster list $S_{clust}$
    if $v_q \in S_{HE}$ then
        Set $\{C_1, \ldots, C_r\} = \arg\max_{C \in S_{clust}} |C \cap S_{v_q}|$.
        if $r > 1$ then $C' = rand\{C_1, \ldots, C_r\}$
        else $C' = C_1$
        $S_{v_q} = S_{v_q} \backslash (\cup_{C \in S_{clust}} (C \cap S_{v_q}) \backslash C')$.
    if $|S_{v_q} \cap (S_{HE} \backslash v_q)| > 1$ then
    ▷ $S_{v_q}$ contains multiple high entropy nodes beside $v_q$
        Among nodes in $S_{HE} \backslash v_q$, keep only the node(s) which have the highest transition probability from $v_q$
        ▷ nodes not in $S_{HE}$ are not affected
    return $S_{v_q}$
```

other extreme, if γ is large, there may be very few or no value around $1/|V|$, resulting in mostly two clusters. In our implementation, we use the Python scikit-learn agglomerative clustering to look for this initial set of (up to) three clusters S_{ini,v_q}. Among these clusters, we choose the cluster with the highest average transition probability from v_q. We define S_{v_q} to be the nodes corresponding to this cluster along with v_q itself since, (i) the constituent nodes have similar probabilities for random walks to end up there when originating from v_q (this follows from the clustering of the probability values), and crucially, these nodes are considered to comprise the immediate local community because (ii) this is the largest (in expectation) such probability.

We consider the absorption probabilities for a random walker starting at v_q to be absorbed at any of the nodes in S_{v_q}, and define the minimum of these values as σ i.e. $\sigma = \min_{v \in S_{v_q}} \hat{P}_{v_q,v}$. Thus, σ can be understood as an effective threshold (deduced a posteriori) above which the probability of being absorbed in the local cluster of v_q is high enough. Proposition 2 and its corollary below show that if v belongs to the local cluster of v_q, but w should also belong to the local cluster of v, then, provided that the absorption probability $p_{vv'}$ at v is not too large ($p_{vv'} \leq \sigma$), w will also belong to S_{v_q}, together with v.

Proposition 2: The probability $\pi_{uw} = \sum_{t \geq 1} \tilde{p}^{(t)}_{uw} p_{ww'}$ to start at u and to be absorbed at $w \neq u$ over time is lower bounded by:

$$\tilde{p}_{uw} p_{ww'} + \pi_{uv} \frac{\pi_{vw}}{p_{vv'}},$$

Proof: We have that $\pi_{uw} = \sum_{t \geq 1} \tilde{p}^{(t)}_{uw} p_{ww'}$ is the probability to start at u and to be absorbed at w over time. We start again with (6):

$$\begin{aligned}
&\pi_{uw} \\
&= (\tilde{p}_{uw} + \sum_{t \geq 2} \sum_{i=1}^{t-1} \tilde{p}^{(i)}_{uv} \tilde{p}^{(t-i)}_{vw} + \sum_{t \geq 2} \sum_{i=1}^{t-1} \sum_{y \neq v} \tilde{p}^{(i)}_{uy} \tilde{p}^{(t-i)}_{yw}) p_{ww'} \\
&\geq (\tilde{p}_{uw} + \sum_{t \geq 2} \sum_{i=1}^{t-1} \tilde{p}^{(i)}_{uv} \tilde{p}^{(t-i)}_{vw}) p_{ww'} \\
&= (\tilde{p}_{uw} + \sum_{s \geq 1} \tilde{p}^{(s)}_{vw} \sum_{i \geq 1} \tilde{p}^{(i)}_{uv}) p_{ww'} = \tilde{p}_{uw} p_{ww'} + \frac{\pi_{uv}}{p_{vv'}} \pi_{vw}.
\end{aligned}$$

The above derivation relied on a Cauchy product and by the invocation of Merten's Theorem where $\sum_{i \geq 1} \tilde{p}^{(i)}_{uv} \rightarrow \pi_{uv}$, $\sum_{s \geq 1} \tilde{p}^{(s)}_{vw} \rightarrow \pi_{vw}$, and both sequences $(\tilde{p}^{(i)}_{uv})_i, (\tilde{p}^{(s)}_{vw})_s$ are absolutely converging to 0. ∎

Corollary 2: Given that the probability π_{uv} to start at u and to be absorbed at v is more than a threshold σ, and that the probability to start at v and to be absorbed at w is also more than σ, if $p_{vv'} \leq \sigma$ then $\pi_{uw} \geq \sigma$.

Proof: Suppose $\pi_{uv}, \pi_{vw} \geq \sigma$, then by the above proposition:

$$\pi_{uw} \geq \tilde{p}_{uw} p_{ww'} + \pi_{uv} \frac{\pi_{vw}}{p_{vv'}} \geq \pi_{uv} \frac{\pi_{vw}}{p_{vv'}} \geq \frac{\sigma^2}{p_{vv'}}.$$

So, $p_{vv'} \leq \sigma$ implies $\pi_{uw} \geq \sigma$. Note that this reasoning is iterative, namely if we now consider that from u, we got absorbed at w, but also from w, we got absorbed at y, with $\pi_{yw} \geq \sigma$, then

$$\pi_{uy} \geq \tilde{p}_{uy} p_{yy'} + \frac{\sigma^2}{p_{ww'}}$$

and again $p_{ww'} \leq \sigma$ implies $\pi_{uy} \geq \sigma$. ∎

If the query node v_q is a low entropic centrality node, it is expected that once a cluster S_{v_q} is formed, nodes belonging to S_{v_q} are unlikely to be high centrality nodes themselves. Otherwise, v_q would have inherited the influence and it would not be a low centrality node itself. However, if the relative difference between what are considered low or high entropic centrality is not high, then several well connected nodes may get clubbed together in the clustering process, further bringing in many other nodes, coalescing a large group of nodes in the early stage of the clustering. Likewise, if the query node v_q belongs to a pre-designated group of high entropic centrality nodes S_{HE}, then there is a risk that it may inadvertently merge multiple clusters which one may want to be separate. In Figure 10 we illustrate one of these scenarios with a toy example, where an initial cluster seeded at the query node 'A' inadvertently includes high centrality nodes 'X', 'Y', 'Z', which would thwart the possibility to identify

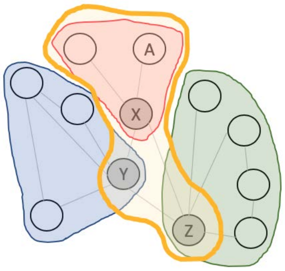


FIGURE 10. Toy example motivating the need for a pruning stage (Algorithm 2).

the desired smaller communities. These motivate the addition of a pruning algorithm described in Algorithm 2, which works as follows.

If the query node v_q is a high entropic centrality node, then we check the intersection of S_{v_q} against existing clusters from S_{clust}, and in case there are non-trivial intersections, and yet there is no unique cluster with which there is a largest intersection, we retain only one of the largest clusters as S_{v_q} chosen randomly, since it otherwise risks merging clusters which ought to be distinct. Otherwise, we retain as S_{v_q} the nodes from the largest intersection, as well as nodes that were in no intersection, but discard the rest. Irrespective of the query node, if there are multiple pre-designated high entropic centrality nodes in S_{v_q} (other than v_q), we retain among these only the ones to which the transition probability from v_q is highest. This pruned list is given to Algorithm 1, where it is checked against existing clusters in S_{clust}, and if there is any intersection, then they are merged, otherwise, S_{v_q} is added as a new cluster in S_{clust}. Note that the pruning mechanism introduces an element of randomization in our algorithm. As such, all the reported results in this paper are based on ten experiment runs. For all but one of the graphs used in our experiments, the conditional statement which introduces the randomization was in fact never triggered, and thus the results are produced in a consistent fashion. Only some variations of the 178 nodes Bitcoin subgraph (results shown in Figure 13) triggered the conditional statement: When it was treated as undirected and unweighted, the conditional statement was triggered but nevertheless resulted in same clusters across the ten experiment runs. For the same graph treated as directed but unweighted, the randomization yielded distinct but very similar cluster results (F-score 0.982 among the distinct results), and one of these result instances is shown.

Having identified localized query-node centric community structures, in the ***second stage***, we agglomerate these to identify clusters at different degrees of granularity. A single stage of agglomeration is almost identical to the initial clustering process described above, with the following subtle changes. The cluster results from the previous step are considered as the new nodes. We still only use the matrix $\hat{P}$, and hence we do not (need to) explicitly define edges connecting the clustered nodes. The new coalesced nodes are assigned an entropic centrality value corresponding to the average of the entropic centrality of their constituent nodes. For transition probabilities across clustered nodes (say $\bar{C}$ and $\tilde{C}$), we considered the minimum, mean and maximum transition probabilities amongst all node pairs $u \in \bar{C}$, $v \in \tilde{C}$ as per $\hat{P}$. Finally, we discard a specific agglomeration in case the resulting agglomerated cluster would not result in a (weakly) connected graph. Our experiments indicate that the best clustering results are obtained using the minimum transition probabilities, as such, we only report the corresponding results in the upcoming subsections.

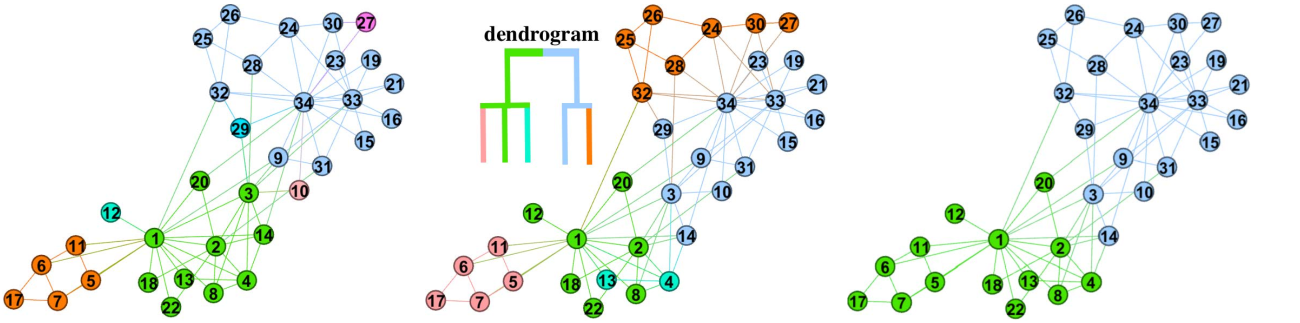
A back of the envelope estimate of the computational complexity of the clustering process is as follows. The computation of $\hat{P}$ is the most expensive (one time) operation, comprising an $O(n^3)$ matrix inversion and an $O(n^2)$ matrix multiplication. Subsequent computation of entropic centrality for a single node is an $O(n)$ summing of a column of the matrix, repeated for all nodes, yielding another $O(n^2)$ operation, making (the first) step numbered '2' in the pseudo-code of Algorithm 1 the most expensive step of the process. Subsequent steps involve operations like sorting, set intersection and merger, clustering of scalar values, for all of which many $O(n^2)$ and lower complexity algorithms exist. Likewise, the agglomerative step uses (significantly) smaller graphs. The absorption probability values for random walkers from all possible query nodes to all destination nodes needs to be stored, which leads to a memory complexity of $O(n^2)$ for the proposed clustering process. In practice, large graphs are sparse. Heuristics for fast, or even distributed computation of matrix inverses, for example [26], [50], can be leveraged to attain significantly reduced computational cost, and improve scalability of our approach.

C. CLUSTERING OF THE KARATE CLUB NETWORK

We consider the karate club network [51] (see Figure 11 for the cluster results), to illustrate and analyze the workings of the clustering algorithm with a toy example with known baseline, before studying larger graphs. In Figure 11b, the initial set S_{ini,v_q} of clusters is shown along with the dendrogram for agglomeration, and Figure 11c shows the final clusters. Clustering obtained using the edge removal technique (20 iterations) from [31] is shown in Figure 11a for comparison. We also show the time evolution of some of the nodes with highest asymptotic entropic centrality, and the node with the lowest asymptotic entropic centrality in Figure 12 (this is in addition to Figure 3 where we demonstrated the time evolution of certain nodes too), to illustrate the behavior of nodes which are bridges, at the interface of the clusters and at the periphery of the network. This top-down clustering approach follows the idea of edge removal from [15], but using the reduction of average Markov entropic centrality to determine which edge to remove.

While it is visually apparent that our approach yields better clustering, we quantify this based on the ground truth [51] using F-score [39]. The result obtained with our approach has a F-score of 0.884 while the one obtained with [31] has a F-score of 0.737.

We furthermore benchmark the computation time: the edge removal based clustering approach [31] took 14.154 seconds, while our final result of 2-clusters were computed in 0.026 seconds. These experiments were run on a 64-bit PC with x64-based Intel(R) Xeon(R) CPU E5-1650 0 @3.20GHz processor and 16 GB RAM. This is easily explained: in our algorithm, the transition probability matrix $\hat{P}$ needs to be computed only once. In contrast, even within a single iteration, the edge removal algorithm [31] needs to recompute the said matrix for every graph instance created by removal of each possible edge, to determine which edge to remove, and this exercise is then repeated in every iteration. That



(a) Edge removal (20 iterations).

(b) Initial clusters (stage 1).

(c) The final 2-cluster result (stage 2).

FIGURE 11. Clustering of the karate club network: using the edge removal clustering of [31] on the left, and the proposed algorithms in the middle (stage 1) and on the right (stage 2, with agglomeration).

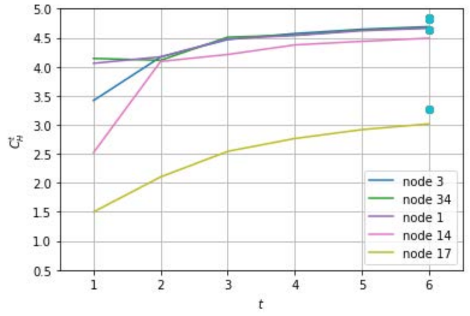


FIGURE 12. Time evolution of entropic centrality of the three nodes with the highest centralities (nodes 3, 34, 1 respectively), node 14, which visually appears in the graph to be at the border of the clusters (and has, in fact, the seventh highest centrality) and the node with the lowest entropic centrality (node 17).

accounts for the huge discrepancy in the computation time, and demonstrates the computational efficacy of our approach.

D. CLUSTERING OF BITCOIN SUBGRAPHS

We apply our proposed clustering algorithm to variations (un/directed, un/weighted) of the 178 node Bitcoin network subgraph [32] and report the results in Figure 13. The results obtained using the edge removal algorithm [31] on the undirected unweighted graph variant is shown on Figure 14. Unless otherwise stated, we use as a parameter $N = 53$, essentially considering S_{HE} to comprise the top 30% entropic centrality nodes (see the sensitivity analysis below).

While it is visually clear from Figure 13 that we obtain different clusters depending on the scenario considered, Table 3 confirms this by reporting F-scores [39] across the graph variants. Also, the effect of the parameter μ (without the transition probabilities being altered by edge weights) is rather low. This reinforces an underlying motivation of our work, namely that which graph variant (un/directed and/or weighted) to study is application dependent, and hence having one graph clustering algorithm that works across all variants is beneficial.

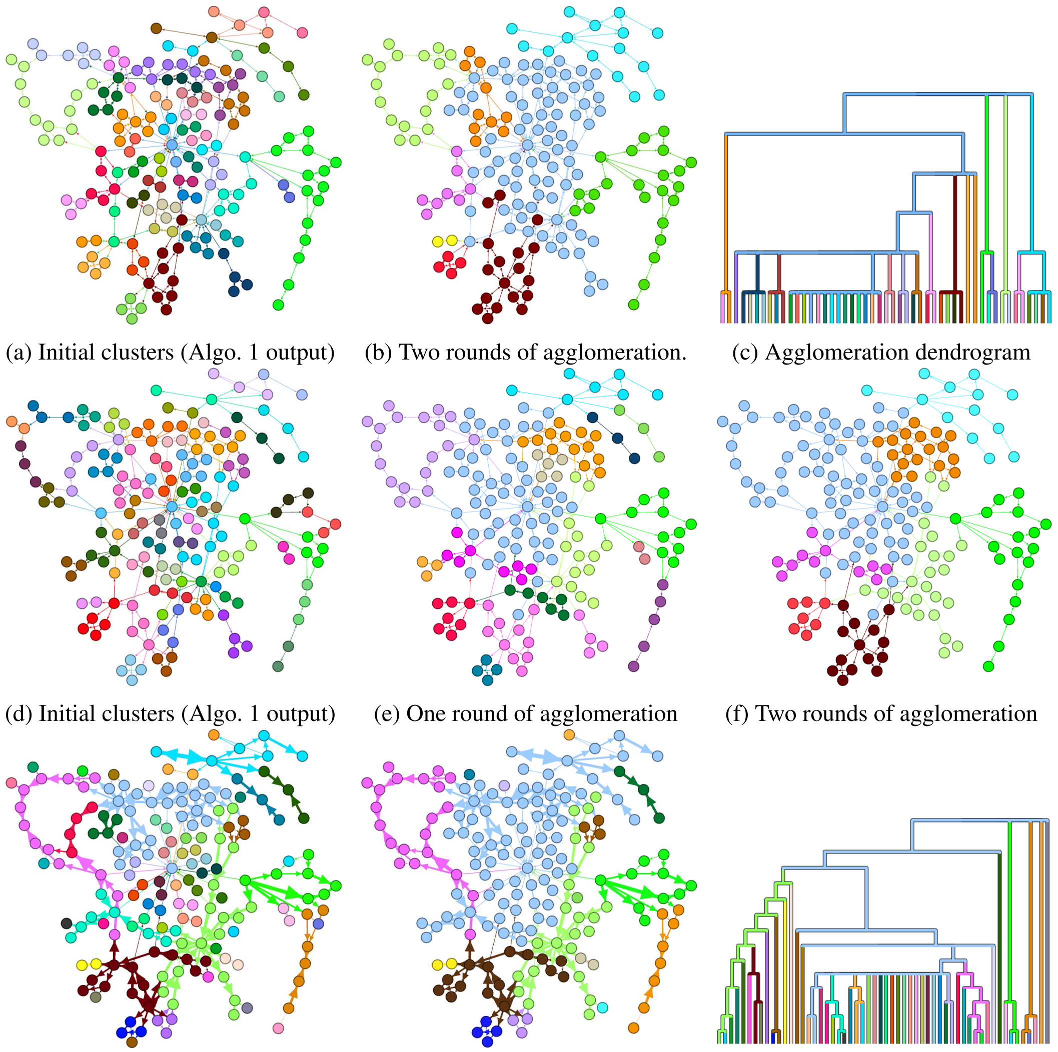
TABLE 3. Pairwise F-score among clusterings achieved for different graph variants. Legend — UU: undirected & unweighted; $D_{x,y}$: Directed with $\alpha = x$, and $\mu = 1$ if $y = 1$, else $\mu = \frac{d_{w,out}(v)}{d_{out}(v)}$ if $y = M$; UU_{er}: edge removal algo [31].

graph	UU	$D_{1,1}$	$D_{w,1}$	$D_{1,M}$	$D_{w,M}$	UU_{er}
UU	1	0.416	0.506	0.372	0.271	0.125
$D_{1,1}$	0.416	1	0.473	0.806	0.364	0.146
$D_{w,1}$	0.506	0.473	1	0.408	0.416	0.122
$D_{1,M}$	0.372	0.806	0.408	1	0.413	0.145
$D_{w,M}$	0.271	0.346	0.416	0.413	1	0.131
UU_{er}	0.125	0.146	0.122	0.145	0.131	1

The clusterings of the undirected unweighted graph found by our algorithm in 1.072 seconds (Figure 13) and by the edge removal algorithm in 3196.07 seconds (Figure 14) are very different (F-score of 0.125). Our algorithm visually yields (more) meaningful results, though in the absence of a ground truth, this assertion remains subjective. The agglomeration process in our algorithm could have been continued beyond the final result shown here, as indicated by the associated dendrogram. The edge removal based clustering was stopped at the 50th iteration, after which no further average entropy reduction was observed with any single edge removal.

When to stop agglomerating is typically purpose dependent and as such, often determined as per users' discretion (see, e.g., [8] for more discussions on cluster validation). The second row of Figure 13 shows various stages of agglomeration, the dendrogram for agglomeration is given for the 1st and 3rd row. There are many community structures that repeat across the considered scenarios, and most of the cluster boundaries can be traced back to the high entropy nodes (Figure 7), yet there are also subtle differences, e.g., in the weighted directed graph, there are instances of single isolated nodes, which stay isolated for several iterations of agglomeration because of weak (low weight) connections.

Sensitivity analysis. For each graph variant, the size of S_{HE} was varied to comprise 10%, 20%, 30% and 40% of the top entropic centrality nodes. For the unweighted and weighted directed graph, the F-score between clusterings



(a) Initial clusters (Algo. 1 output) (b) Two rounds of agglomeration. (c) Agglomeration dendrogram

(d) Initial clusters (Algo. 1 output) (e) One round of agglomeration (f) Two rounds of agglomeration

(g) Initial clusters (Algo. 1 output) (h) Four rounds of agglomeration (i) Agglomeration dendrogram

FIGURE 13. Clustering of the 178 node bitcoin subgraph (results obtained in 1.072, 1.077 and 1.123 seconds respectively): 1st row - undirected unweighted graph, 2nd row - directed unweighted graph, 3rd row - directed weighted graph ($\alpha = 1$, $\mu = \frac{d_{w,out}(v)}{d_{out}(v)}$).

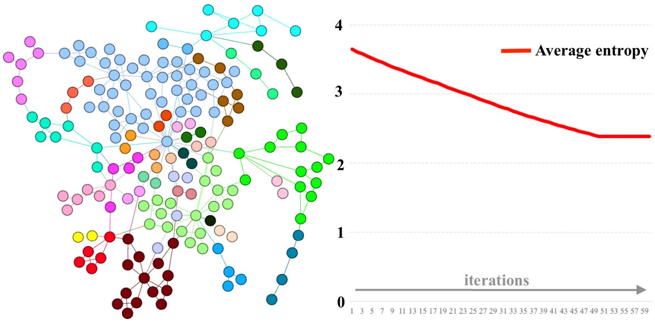


FIGURE 14. Clustering by iterative edge removal [31].

obtained with 10% and the others are 0.824 and 0.989 respectively, while all the other pairs have F-score of 1. This suggests very consistent results in these cases, irrespectively of the choice of $|S_{HE}|$. However for the unweighted undirected graph, $|S_{HE}|$ has a significant impact, with the F-score between 20% and 40% being the lowest at 0.626, while the

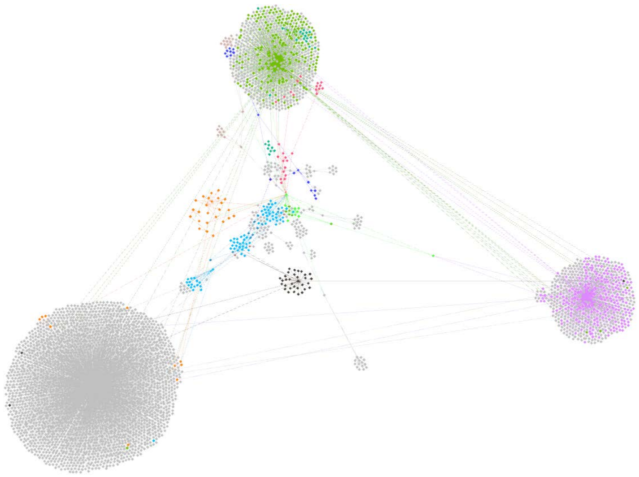


FIGURE 15. Clustering of a Bitcoin subgraph involved in the Ashley-Madison extortion scam [41].

best score of 0.912 is obtained between 30% and 40%. This justifies the use of the top 30% entropic centrality nodes for the reported results. Looking back at Figure 9, we observe that considering only 20% for the size of S_{HE} means including a number of query nodes with relatively low highest probabilities, while between 30% and 40%, these highest probabilities are not changing significantly, which is consistent with the observed variations in sensitivity.

We next consider another Bitcoin subgraph, but this time, we use a specific subgraph [33] of 4571 nodes, constructed around Bitcoin addresses suspected to be involved in the Ashley-Madison extortion scam [41]. The result of the clustering algorithm is shown on Figure 15 and Figure 16: (1) the graph was considered once unweighted (the emphasis is thus on node connections), once weighted (to capture the amount of Bitcoins involved), (2) the asymptotic Markov entropic centrality was used (we have no specific diameter of interest), (3) the top 30% nodes with the highest centrality are chosen as high centrality nodes S_{HE}, (4) five agglomeration iterations were performed for clustering the unweighted graph, and one for the weighted variant.

On Figure 15, showing the overall unweighted graph, there are three visually obvious main clusters: the upper green cluster, the purple cluster on the right, and the grey cluster on the left. The first observation is that the grey color here only represents nodes whose cluster size is too small to be significant (only 5 iterations were performed), they are thus kept in grey so as to make the other clusters more visible. The green and purple clusters are easily interpreted: each contains one Bitcoin address that is a hub for all its neighbors.

We then zoom into the central clusters, shown on Figure 16a. The actual relationship among the constituent wallet addresses in a cluster can be determined e.g. by using `blockchain.com/explorer`. We observe a green cluster near the middle (boxed). In our layout, we have isolated one of the constituent nodes (on the right, encircled), to show that the nodes in this collection have multiple mutual connections, as expected among nodes within a cluster. We see that the encircled node above the boxed group has also been assigned to the same cluster. This node is in fact connected to several of the other clusters that have been identified with our algorithm, and is one of the high centrality nodes, which lies at the interface of clusters. It happens to have been assigned to the green cluster, since each node is assigned to at most a single cluster. Some of the nodes in the (boxed) cluster were previously identified to be suspect addresses involved in the Ashley-Madison data breach extortion scam [41]. The resulting clusters thus help draw our attention to the other nodes which have been grouped together, since it indicates that Bitcoin flows have circulated among them, for their relationship with the already known suspected nodes to be investigated further.

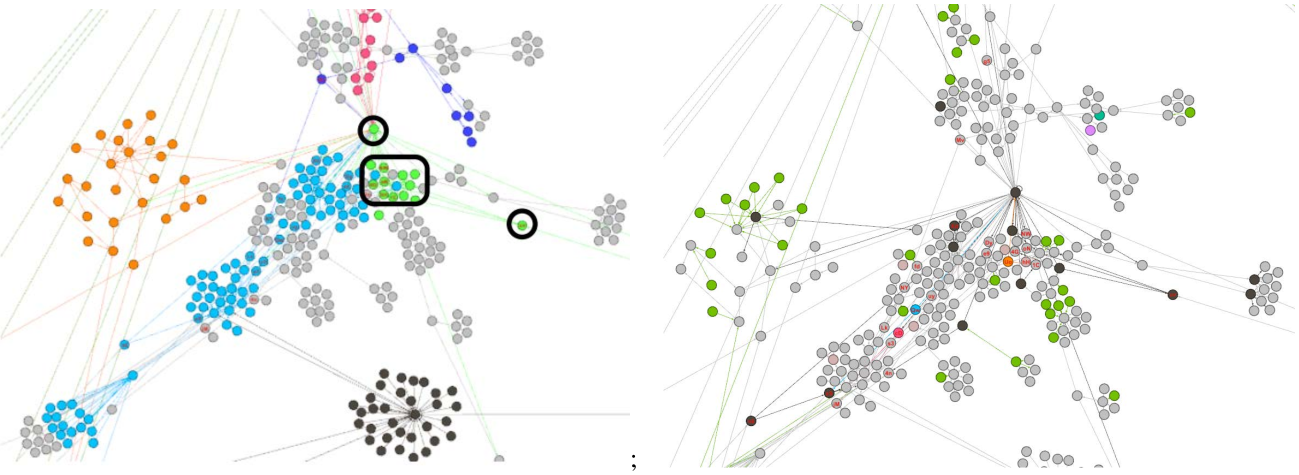
Zooming into the weighted graph gives a very different picture: since the amounts of Bitcoin involved drive the clustering in this case, we prominently see two clusters highlighting nodes dealing with high volume of Bitcoins. This confirms an expected behavior from scammers, which consists of collecting few Bitcoins from many addresses to avoid attention. Combining both clustering results however correlate nodes that are likely to be involved in the scam, together with nodes dealing with high volume of Bitcoins. For example, this could be a direction to consider for Bitcoin forensics: nodes appearing in clusters by interpreting the graph in both manners could possibly be involved in aggregating scam money, since they stand out both in terms of the volume of Bitcoin they handle, and in terms of the frequency of interactions with multiple suspected wallet addresses.

E. BENCHMARKING WITH SYNTHETIC GRAPHS

In the previous subsection, we looked at the clusterings that the algorithm provides with Bitcoin subgraphs, for which no ground truth is available. In order to benchmark our proposed algorithm rigorously, we thus further experiment with graphs for which some form of ground truth is assumed.

An acknowledged concern in the research community is that a unique objective benchmark to compare graph clustering algorithms is not feasible [38], [49]. Different algorithms may yield different clusters for a given graph, that may each be meaningfully interpreted based on distinct contexts. Conversely, in the real world, connections may have been induced as a consequence of multiple causes (contexts), and using the meta-data representing a subset of these contexts to determine a 'ground truth' for the resulting graph may not be accurate. Furthermore, there may be implicit or explicit hierarchical community structures in the graph, or the communities may be fuzzy, and a clustering algorithm may find clusters at a coarser or finer granularity than the one considered as the ground truth.

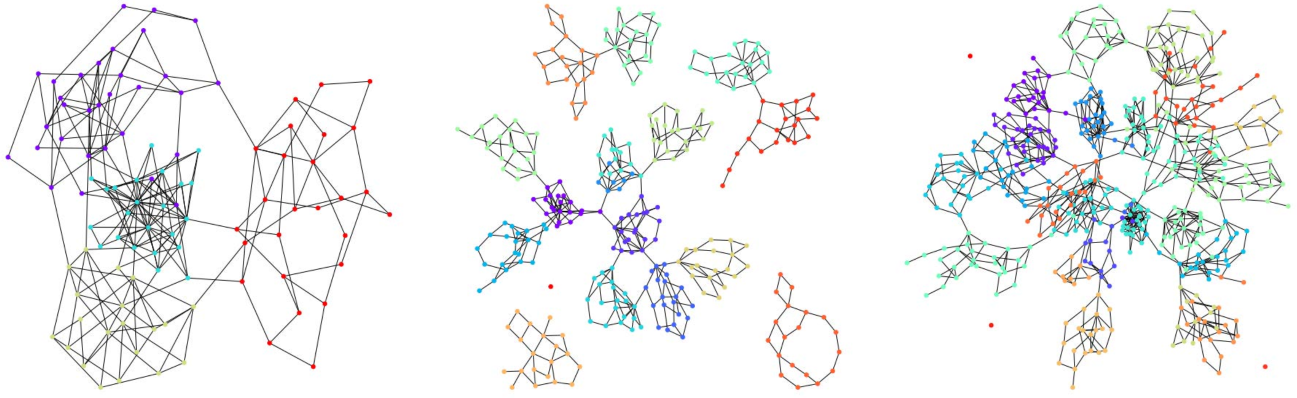
Synthetic graphs (e.g [23]) are considered to alleviate the issue of the lack of a unique and objective ground truth. Yet such synthetic graphs may not carry all the characteristics and associated complications of a real network.



(a) Graph treated as unweighted

(b) Graph treated as weighted

FIGURE 16. **Zoom in of the Ashley-Madison extortion scam (Bitcoin transactions induced) graph.**



(a) 100 nodes graph: 4 clusters detected with F-score 0.900

(b) 300 nodes graph: 15 clusters detected with F-score 0.973

(c) 500 nodes graph: 21 clusters detected with F-score 0.909

FIGURE 17. **Clusterings of synthetically generated networks with mixing probability $\mu = 0.1$.**

Considering the merits of benchmarking with graphs with ground truth, but also the inherent limitations associated with any particular instance(s) of real or synthetic (family of) graph(s), we study a wide range of graphs with assumed ground truth. We next discuss our experiments with synthetic graphs, before studying some real graphs in the following subsection.

The principal idea of generating synthetic graphs with a known ground truth is to first create isolated subgraphs (with certain properties such as a given node degree distribution) that represent the ground truth communities. Then, rewiring of a fraction of the connections is carried out to establish cross community links, such that, probabilistically, a $1 - \mu$ fraction of links are with nodes within the same community, while a fraction μ (mixing probability) of connections are with other nodes. Though this rewiring process itself might affect the neighborhood of individual nodes to an extent that it materially changes the community it actually belongs to (particularly for high values of the mixing probability μ), the original allocation of the communities is considered to continue to hold, and is treated as the ground truth. For the reported experiments below, we used synthetic benchmark graph instances randomly generated using NetworkX.

For sanity check and to manually (visually) interpret the results, we start the benchmarking with small graphs and a small value of $\mu = 0.1$. The clusters identified by our algorithm for synthetic graphs with 100, 300 and 500 nodes are shown in Figure 17 for $\mu = 0.1$. Visually, we see that the algorithm yields meaningful clusters. We also determine the F-score with respect to the ground truth as determined by the network generation process, and across the different networks we observe very good (0.9 or above) F-score values. For the 100 nodes graph, one can visually see certain nodes, particularly in the middle group being allocated cluster

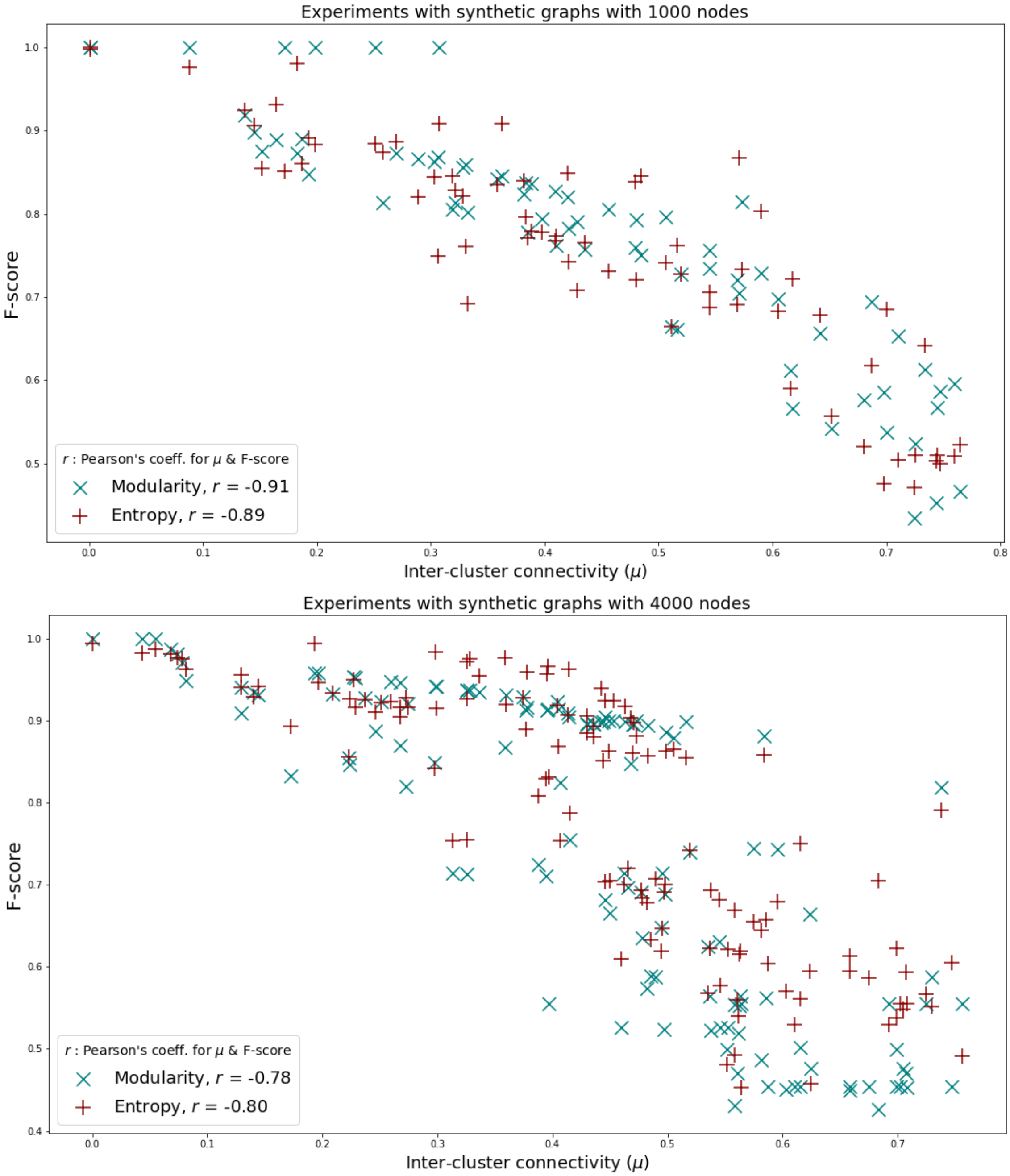
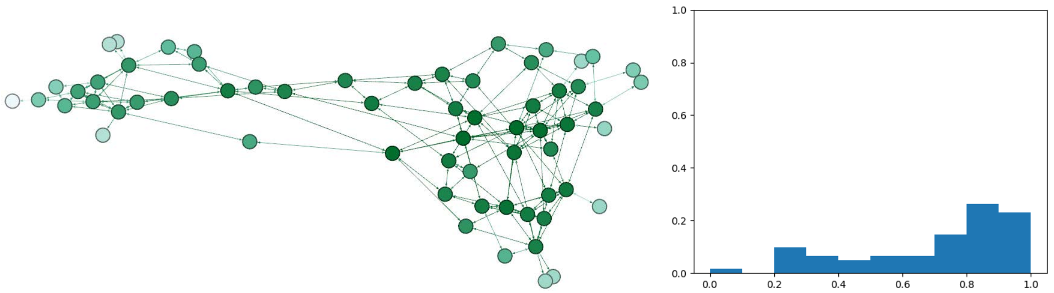


FIGURE 18. Scatter plots of F-scores for large-scale benchmarking of our Entropy based graph clustering approach with randomly generated synthetic graphs comprising 1000 and 4000 nodes respectively for a wide range of inter-cluster linkage μ characteristics are shown. Modularity optimization based Louvain [5] community detection algorithm is also shown to provide a point of comparison.

labels that mismatch, explaining the relatively lower score of 0.9 among these graph instances. For the 500 nodes graph instance, the isolated nodes had distinct labels in the ground truth. Some other misattributions can also be seen visually, explaining the relatively lower score of 0.909. The 300 nodes graph instance had disconnected components, which might have made it easier to identify relevant communities, yielding a noticeably high score of 0.973.

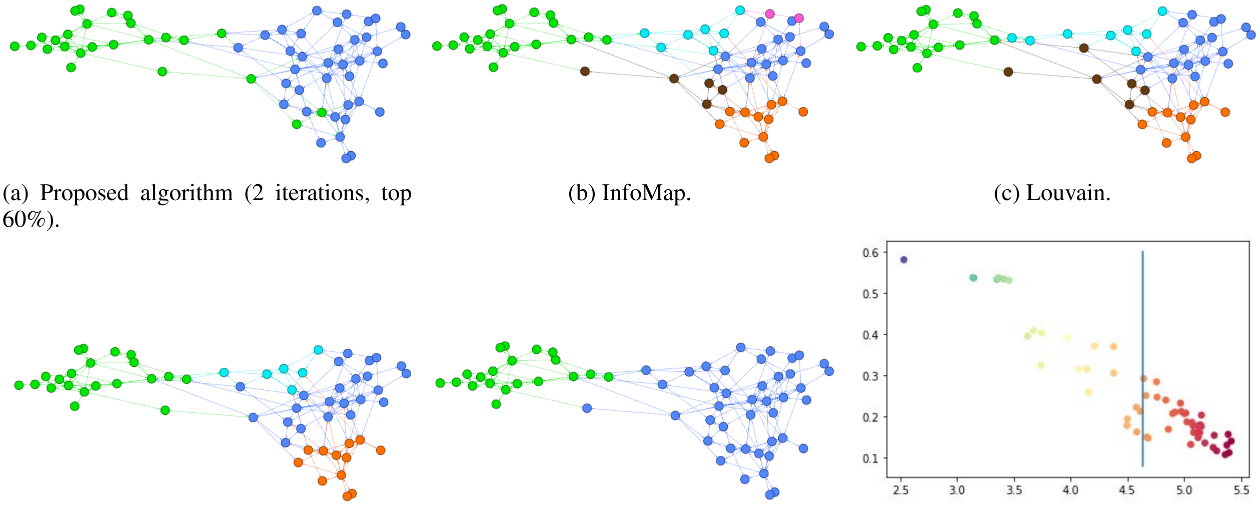
We next extend our study with larger scale experiments both (i) in terms of graph size (1000 and 4000 nodes) and (ii) in terms of range of μ values representing different extents of cross-community linkages. We show the scatter plots of observed F-scores for our **Entropy** based graph clustering algorithm. For a point of reference, we also provide the results observed for clustering with Modularity optimization [5]. For relatively smaller values of μ, perfect clustering is obtained



(a) The asymptotic Markov entropic centralities (darker the color, higher the relative entropic centrality score).

(b) Histogram of the normalized (by the maximum) centralities.

FIGURE 19. The dolphin network.



(a) Proposed algorithm (2 iterations, top 60%).

(b) InfoMap.

(c) Louvain.

(d) Label propagation.

(e) The ground truth [28].

(f) Maximum absorption probability as a function of the entropic centrality.

FIGURE 20. Clusterings of the dolphin network.

by [5] while near perfect clustering is also obtained by our approach. For the rest of the spectrum of μ values, performance of both the approaches varies to certain extent (more so in the larger graphs with 4000 nodes), but overall both algorithms yield good results, e.g., F-score is consistently higher than 0.8 for $\mu < 0.3$, and the deterioration of the F-score with increasing μ is gradual, rather than sharp. Furthermore, while the performance varies across different graph instances, very high (absolute values of) Pearson's correlation coefficients between F-score and μ (precise correlation coefficient r values are indicated in the figures) indicate a good degree of consistency in the behaviour for both the algorithms. From individual data points, we observe that among the two algorithms, there is no clear winner, and each outperforms the other for some graph instances across most of the μ value ranges. These large-scale benchmarking experiments with randomly generated synthetic graphs demonstrate the efficacy of our proposed approach on its own. While the original objective of our proposal was to investigate a new way to carry out graph clustering rather than to necessarily compete with existing approaches, the comparison using synthetic graphs with one of the popular existing approaches further demonstrates that the quality of clustering results obtained by our approach is in fact comparable.

F. BENCHMARKING WITH REAL WORLD GRAPHS

Since synthetic graphs may not exhibit all the nuances of communities occurring in the real world, we complement our study with benchmarking experiments using networks with known ground truth, namely, the dolphin network [28] and the American college football network [15] which were previously used in the work [31] that we extend. Moreover, we extend the comparative aspect of our evaluation, and to that end we compare our approach with other popular community detection algorithms, namely, InfoMap [44], label propagation [2] and Louvain clustering [5].

1) CLUSTERING OF THE DOLPHIN NETWORK

We consider the dolphin network [28], an undirected unweighted social network where bottlenose dolphins are represented as nodes and association between dolphin pairs are represented as links. The network comprises 62 nodes and 159 links, and it was noticed that the dolphin community splits into two communities [28] comprising 20 nodes and 42 nodes. We use this as the ground truth.

In Figure 19a, we show the network structure and the corresponding relative (that is, normalized by the maximum observed value) Markov entropic centralities: the darker the node color, the higher the relative centrality. Figures 19b and 20f respectively depict the relative Markov entropic centrality (fractional, bucketed) distribution and the scatter diagram of the absolute entropic centrality score (x-axis), versus the maximum absorption probability at any node (y-axis) for a random walker starting from the given node.

The histogram indicates that it would be more meaningful to choose the clustering parameter S_{HE} to include the top 50%-70% (instead of $\approx$ 30%, as used in the previous experiments) since more than 60% of the nodes have normalized entropy value above 0.7. The scatter diagram helps us see that, indeed, taking the top 30% nodes for S_{HE} would mean including many nodes whose highest probability is relatively small, while there are few such nodes for the threshold at 60% (the vertical line demarcates the top 60% nodes on the right). The result obtained with two iterations of the algorithm is shown on Figure 20a, next to clustering results obtained using InfoMap [44], Louvain [5] and label propagation [2] (with their default parameters). We observe visually that the proposed clustering produces a better result compared to other clusterings. This is confirmed by computing the F-score [39] for each clustering result against the ground truth: the F-score of the proposed algorithm is 0.858, in contrast, it is 0.545 for InfoMap, 0.565 for Louvain, and 0.657 for label propagation. These three community detection techniques also find more than two clusters. From Figure 20, we also visually infer that the other clustering results could be improved if agglomeration techniques were applied to the smaller communities located on right hand side of the dolphin network. In fact, it could be argued that even though the group of dolphins split in two groups (which is the basis of the ground truth), it does not preclude the existence of further smaller communities within those two split groups, which could then be what is being detected by these algorithms.

2) CLUSTERING OF THE AMERICAN COLLEGE FOOTBALL NETWORK

We next consider the American college football network [15], an undirected unweighted network representing the Division-I football games from Fall 2000. A team is represented as a node, and a game between two teams is represented as a link between two nodes. There were in total 115 teams and 613 games. Teams were divided into 12 conferences, and teams in the same conference frequently had games against others. We treat the 12 conferences as the network's ground truth, comprising 12 clusters.

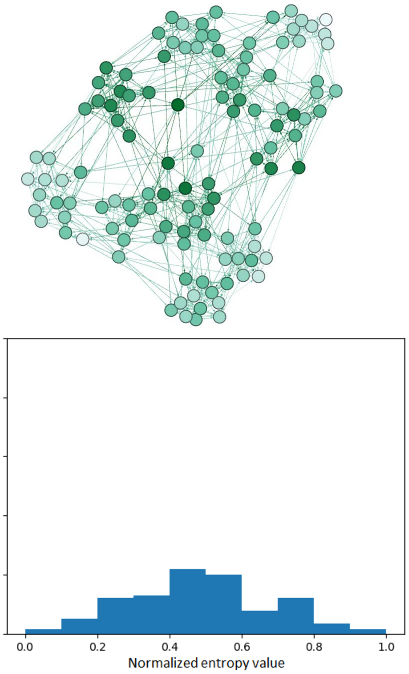
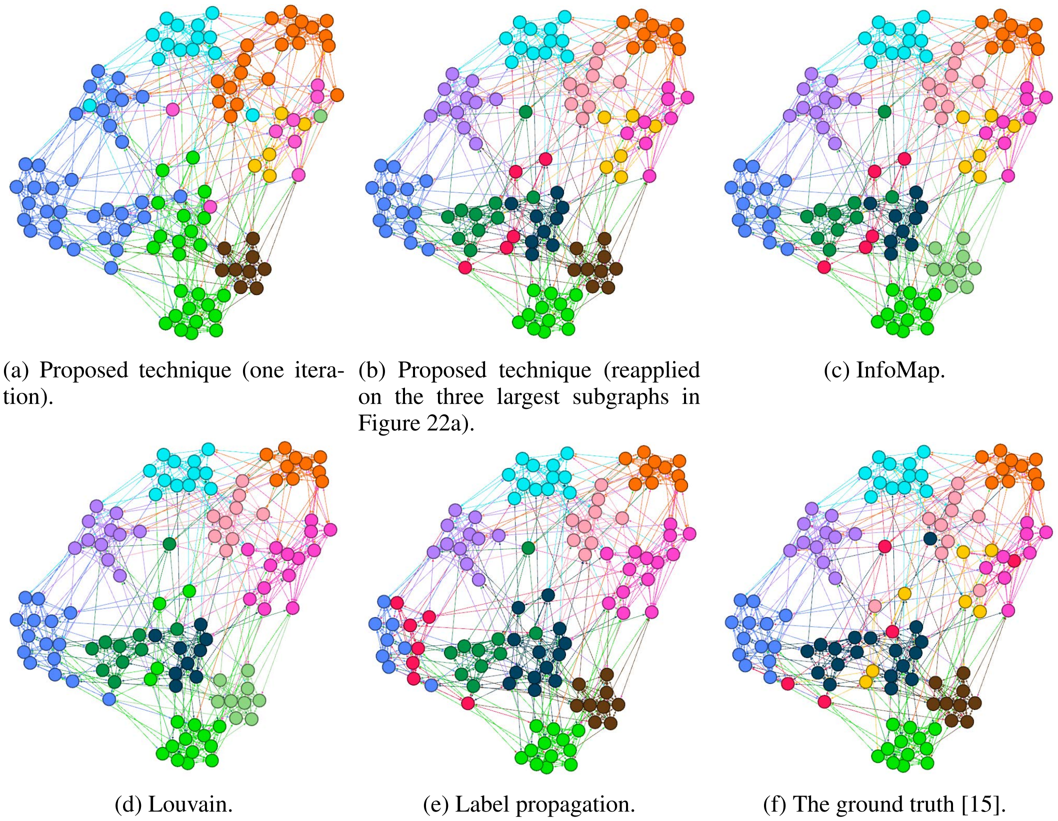


FIGURE 21. The American College football network [15] is shown (darker the color, higher the relative entropic centrality score). The histogram of the normalized (by the maximum) centralities distribution for the same network is also shown.

Figure 21 shows the network and the corresponding, relative (that is, normalized by the maximum entropic centrality value), Markov entropic centralities: darker the color, higher the relative entropic centrality score. We observe that a majority of nodes have normalized entropic centrality between 0.2 to 0.4. This helps us to identify our clustering parameter S_{HE} to determine the set of high entropy nodes deemed as center/border of a cluster. Accordingly, we chose S_{HE} to comprise the top 50/60/70/80% entropic centrality nodes. We registered F-scores against ground truth as 0.273, 0.406, 0.409, and 0.517 respectively. The scatter diagram showing the (absolute) entropic centrality and the maximum absorption probability at any node for a random walker starting at corresponding nodes is shown in Figure 23 (left). The vertical line in the diagram shows the demarcation for S_{HE} for 80%. Unlike for the dolphin network, there is barely any node with distinctively high probability. The threshold of 80% separates a few nodes with both slightly highest entropic centrality and highest probability. The result with F-score 0.517 is shown in Figure 22a. We stop at the first iteration since our clustering technique is a bottom-up approach, which means that second iteration will produce fewer number of clusters. Based on the ground truth (Fig. 22f), we notice for our algorithm a similar behavior as was observed with the other algorithms for the dolphin network, namely: the algorithm coalesced several of the ground truth communities to create larger communities. By extracting the largest three subgraphs of these


(a) Proposed technique (one iteration).

(b) Proposed technique (reapplied on the three largest subgraphs in Figure 22a).

(c) InfoMap.

(d) Louvain.

(e) Label propagation.

(f) The ground truth [15].

FIGURE 22. Clusterings of the American College football network.

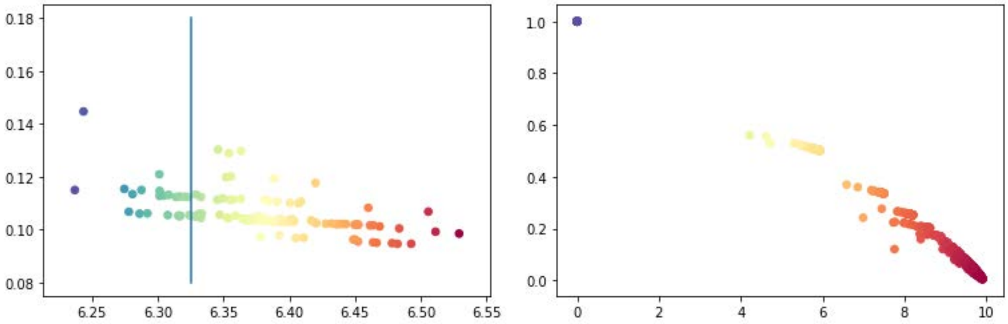


FIGURE 23. Scatter diagram with the entropic centrality of nodes on the *x*-axis, and on the *y*-axis, the maximum absorption probability at any node for a random walker starting at that node: the American college football network (left), and the EU mail network (right).

larger communities and re-applying the algorithm on each of these subgraphs (again with parameter S_{HE} consisting of the top 80% entropic centrality nodes), we obtained a new group of clusters, shown on Fig. 22b, with a significantly improved F-score of 0.811. The overall computation time was a total of 0.221 seconds. We compare this with results obtained with InfoMap [44] (F-score of 0.904 and total time of 0.013 seconds), Louvain [5] (F-score of 0.823 and total time of 0.002 seconds), and label propagation [2] (F-score of 0.796 and total time of 0.001 seconds) as shown in Figure 22 along with ground truth. In the ground truth [15], a cluster with yellow color and another cluster with crimson color spread their members out over the network. Considering this anomalous ‘ground truth’, ours, as well as other clustering techniques, produce very good results. InfoMap has the highest F-score, Louvain has similar F-score to our clustering technique, and label propagation has the lowest F-score.

Finally, we considered the European email network [25] representing email communication in a large European research institution, among members that belong to 42 departments (thus 42 clusters). The corresponding scatter diagram is shown on the right of Figure 23. Looking at the scatter diagram from left to right, we find a few nodes with entropy 0. These are isolated nodes with no edge. Then there is a small group with entropies varying between 4 and 6, and finally, on the right, the bulk of nodes have entropies more than 6 and highest probability less than 0.3. This suggests that the proposed algorithm will have difficulties in identifying clusters, since choosing S_{HE} to include the large right group gives too many clusters, but either iterating or taking smaller S_{HE} leads to too few clusters. This demonstrates how the scatter diagram informs whether and when our approach is suitable for clustering a given graph instance.

V. CONCLUDING REMARKS

In this paper, we investigated the entropic centrality of a graph, using the spread/uncertainty of a random walker's eventual destination as a measure, that is applicable for all families of graphs: un/weighted, un/directed. Studying the probability distribution of a random walker to be absorbed at any given node when initiated at a node of a given entropic centrality, we established principled insights on how to choose query nodes for random walkers, and how to exploit said probability distribution to identify local community structures. We utilized these mechanisms to realize heuristic bottom-up clustering algorithms, relying on the centrality informed choice of query nodes, which inherit the universality of the entropic centrality model. Thus, it is also applicable across families of graphs. We benchmarked the proposed clustering mechanism using a variety of data sets, and by comparing it with other popular algorithms. Given the principles that guided the design of our heuristics, we also explored how the underlying analysis could be leveraged to reason about when and whether our algorithm is suitable to cluster a given graph.

Given the bottom-up and localized nature of the most relevant information that are used in the decision making process, in the future, we want to explore the trade-offs in the quality of results obtained against computational scalability and possible distribution/parallelization, if partial information is used to compute approximate centrality scores and random walker distributions.

Our model naturally fits applications such as the flow of money and its confined circulation among subsets of users, where the volume or frequency of interactions can be mapped to edge weights, and the direction of the flow is vital. Money laundering and cryptocurrency forensics are thus application areas of interest, which we want to explore in the immediate future with the designed tools.

REFERENCES

[1] M. Alamgir and U. von Luxburg, "Multi-agent random walks for local clustering on graphs," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 18–27.

[2] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using PageRank vectors," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2006, pp. 475–486.

[3] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Proc. AAAI Conf. Weblogs Social Media*, 2009, pp. 361–362.

[4] Y. Bian, J. Ni, W. Cheng, and X. Zhang, "Many heads are better than one: Local community detection by the multi-Walker chain," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 21–30.

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[6] S. P. Borgatti, "Centrality and network flow," *Social Netw.*, vol. 27, no. 1, pp. 55–71, Jan. 2005.

[7] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation," *Social Netw.*, vol. 30, no. 2, pp. 136–145, May 2008.

[8] G. Brock, V. Pihur, S. Datta, and D. Datta, "Clvalid: An R package for cluster validation," *J. Stat. Softw.*, vol. 25, no. 4, pp. 1–22, 2018.

[9] D. Bucur, "Top influencers can be identified universally by combining classical centralities," *Sci. Rep.*, vol. 10, no. 1, pp. 1–14, Dec. 2020.

[10] H. Cherifi, G. Palla, B. K. Szymanski, and X. Lu, "On community structure in complex networks: Challenges and opportunities," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–35, Dec. 2019.

[11] J. Coutinho. (2017). *Ucinet Software: Cocaine Dealing Natarajan*. [Online]. Available: https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/cocaine-dealing-natarajan

[12] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, Jan. 1978.

[13] Z. Ghalmane, M. E. Hassouni, and H. Cherifi, "Immunization of networks with non-overlapping community structure," *Social Netw. Anal. Mining*, vol. 9, no. 1, pp. 1–22, Dec. 2019.

[14] Z. Ghalmane, C. Cherifi, H. Cherifi, and M. E. Hassouni, "Centrality in complex networks with overlapping community structure," *Sci. Rep.*, vol. 9, no. 1, pp. 1–29, Dec. 2019.

[15] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[16] K.-I. Goh, B. Kahng, and D. Kim, "Universal behavior of load distribution in scale-free networks," *Phys. Rev. Lett.*, vol. 87, no. 27, Dec. 2001, Art. no. 278701.

[17] S. Guiasu, "Weighted entropy," *Rep. Math. Phys.*, vol. 2, no. 3, pp. 165–179, 1971.

[18] N. Gupta, A. Singh, and H. Cherifi, "Centrality measures for networks with community structure," *Phys. A, Stat. Mech. Appl.*, vol. 452, pp. 46–59, Jun. 2016.

[19] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf. (SciPy)*, 2008, pp. 11–15.

[20] Y. Kim, S.-W. Son, and H. Jeong, "Finding communities in directed networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, no. 1, Jan. 2010, Art. no. 016103.

[21] M. Kumar, A. Singh, and H. Cherifi, "An efficient immunization strategy using overlapping nodes and its neighborhoods," in *Proc. Companion Web Conf. Web Conf. (WWW)*, 2018, pp. 1269–1275.

[22] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, Nov. 2009, Art. no. 056117.

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.

[24] C. Lee and P. Cunningham, "Community detection: Effective evaluation on large social networks," *J. Complex Netw.*, vol. 2, no. 1, pp. 19–37, Mar. 2014.

[25] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discovery From Data*, vol. 1, no. 1, p. 2, Mar. 2007.

[26] S. Li, *Fast Algorithms for Sparse Matrix Inverse Computations*. Stanford, CA, USA: Stanford Univ., 2009.

[27] S.-L. Luo, K. Gong, and L. Kang, "Identifying influential spreaders of epidemics on community networks," 2016, *arXiv:1601.07700*.

[28] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, Sep. 2003.

[29] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Phys. Rep.*, vol. 533, no. 4, pp. 95–142, Dec. 2013.

[30] N. Meghanathan, "Neighborhood-based bridge node centrality tuple for complex network analysis," *Appl. Netw. Sci.*, vol. 6, no. 1, pp. 1–36, Dec. 2021.

[31] A. G. Nikolaev, R. Razib, and A. Kucheriya, "On efficient use of entropy centrality for social network analysis and community detection," *Social Netw.*, vol. 40, pp. 154–162, Jan. 2015.

[32] F. Oggier, S. Phetsouvanh, and A. Datta, *A 178 Node Directed Bitcoin Address Subgraph*, document DR-NTU (Data), V1, 2018, doi: 10.21979/N9/TJMQ8L.

[33] F. Oggier, S. Phetsouvanh, and A. Datta, *A 178 Node Directed Bitcoin Address Subgraph*, document DR-NTU (Data), V1, 2018, doi: 10.21979/N9/TJMQ8L.

[34] F. Oggier, S. Phetsouvanh, and A. Datta, "Entropic centrality for non-atomic flow networks," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Oct. 2018, pp. 50–54.

[35] F. Oggier, S. Phetsouvanh, and A. Datta, "A split-and-transfer flow based entropic centrality," *PeerJ Comput. Sci.*, vol. 5, p. e220, Sep. 2019.

[36] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Netw.*, vol. 32, no. 3, pp. 245–251, Jul. 2010.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, and M. E. P. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[38] L. Peel, D. B. Larremore, and A. Clauset, "The ground truth about metadata and community detection in networks," *Sci. Adv.*, vol. 3, no. 5, May 2017, Art. no. e1602548.

[39] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowl. Inf. Syst.*, vol. 19, no. 3, pp. 361–394, Jun. 2009.

[40] S. Phetsouvanh, "Graph analysis techniques and applications to bitcoin forensics," Ph.D. dissertation, School Comput. Sci. Eng., NTU, Singapore, 2019.

[41] S. Phetsouvanh, F. Oggier, and A. Datta, "EGRET: Extortion graph exploration techniques in the bitcoin network," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 244–251.

[42] R. Quax, A. Apolloni, and P. M. A. Sloot, "Towards understanding the behavior of physical systems using information theory," *Eur. Phys. J. Special Topics*, vol. 222, no. 6, pp. 1389–1401, Sep. 2013.

[43] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, "Characterizing the interactions between classical and community-aware centrality measures in complex networks," *Sci. Rep.*, vol. 11, no. 1, pp. 1–15, Dec. 2021.

[44] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[45] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev. I*, vol. 1, no. 1, pp. 27–64, 2007.

[46] C. Sciarra, G. Chiarotti, F. Laio, and L. Ridolfi, "A change of perspective in network centrality," *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, Dec. 2018.

[47] M. M. Tulu, R. Hou, and T. Younas, "Identifying influential nodes based on community structure to speed up the dissemination of information in complex network," *IEEE Access*, vol. 6, pp. 7390–7401, 2018.

[48] F. Tutzauer, "Entropy as a measure of centrality in networks characterized by path-transfer flow," *Social Netw.*, vol. 29, no. 2, pp. 249–265, May 2007.

[49] L. U. von and R. I. W. Guyon, "Clustering: Science or art," in *Proc. Workshop Unsupervised Transf. Learn.*, 2012, pp. 65–79.

[50] J. Xiang, H. Meng, and A. Aboulnaga, "Scalable matrix inversion using MapReduce," in *Proc. 23rd Int. Symp. High-Perform. Parallel Distrib. Comput. (HPDC)*, 2014, pp. 177–190.

[51] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropolog. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[52] Z. Zhao, X. Wang, W. Zhang, and Z. Zhu, "A community-based approach to identifying influential spreaders," *Entropy*, vol. 17, no. 4, pp. 2228–2252, 2015.

FRÉDÉRIQUE OGGIER is currently an Associate Professor with the Division of Mathematical Sciences, Nanyang Technological University, Singapore. Her research interests include algebra and number theory and their applications to coding theory and security.

SILIVANXAY PHETSOUVANH received the B.Eng. degree in computer engineering and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014 and 2019, respectively. When he was a Ph.D. student, his work encompassed the topics of graph analysis techniques and applications to bitcoin forensics. He is currently a Cloud Architect with the E-Government Center, Ministry of Technology and Communications, Laos.

ANWITAMAN DATTA is currently an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Additionally, he works as a Senior Scientific Officer in a consulting role with QPQ.IO. His research interests include large-scale resilient distributed systems, information security, and applications of data analytics. Presently, he is exploring topics at the intersection of computer science, public policies & regulations along with the wider societal and (cyber) security impact of technology. This includes the topics of social media and network analysis, privacy, cyber-risk analysis and management, cryptocurrency forensics, the governance of disruptive technologies, and impact and use of disruptive technologies in digital societies and government.