Networks Mini-Project: Comparing node similarities on networks

Blas Kolic

April 9, 2019

Contents

Abstract

Complex networks often show important patterns and relations beyond their modular structure. We try to capture such behaviours by constructing similarity networks and comparing their corresponding community partitions. Such partitions are obtained by optimizing Stability as described in [1] for different resolution parameters. Two kind of node-similarity measures are used for the analysis, namely role-based similarity [2] and dynamical embedding similarity [3] . Similarity networks for each measure are constructed by obtaining the relaxed minimum spanning tree [4] of the similarity matrices. We tests the similarity networks in a scale-free network and a dolphin social network, where differences between role-based and dynamical embedding similarities are discussed for both examples.

Many real-world systems can be abstracted as a network representing their different actors with their interactions. Such systems can go from arrangements of chemicals in a molecule, to the World Wide Web, to people interacting in social media. In the context of graph theory, one can ask questions like "Is there a modular structure on these systems?", "How does information diffuse?", or "How similar is one actor to the other?". The objective of this work is to define quantities to compare such actors based on their interactions, namely similarity measures and graph partition optimisation, and compare such quantities when applied to different graphs, helping to address several aspects of the posed questions.

Throughout the report, we will consider $\mathcal{G} = (V, E)$ to be a graph with $N = |V|$ vertices, $m = |E|$, and A its $N \times N$ adjacency matrix. Section 1 discusses different node-similarity measures for comparing vertices. Section 2 describes how to construct similarity networks with the Relaxed Minimum Spanning Tree algorithm. Section 3 describes the community detection algorithm used for the similarity networks. Section 4 takes some example networks to tests all of the algorithms and compare the similarity measures. Finally, Section 5 discusses the results.

1 Similarity measures

An important aspect of a network is to see how similar is one vertex with each other, and answers can be widely different depending on what we understand by similarity. Are two vertices similar if they share the same role in the network? Are two vertices similar if they share a similar set of neighbours? Are they similar if they spread information throughout the network in the same way? In this section we quantify this by taking two

different approaches: one is Role-Based Similarity [2], which describes the similarity between vertices based on the profile of their in-going and out-going paths. The other one is Dynamical Embedding Similarity [3], that does so by comparing the impulse response of vertices immersed in a dynamical system.

1.1 Role-based similarity

Role-Based Similarity (RBS) is a kind of node-similarity measure that compares the path structure of different vertices, and it's usually used to identify what is the role of the different vertices in a network has. The following description will be encompassed by the work in [2, 4]. Consider a graph with adjacency matrix A. The number of out(in)-going paths of length k for vertex i is given by $\sum_j A^k_{ij}$ ($\sum_j {A^T}^k_{ij}$). Thus, we can construct a matrix comprehending the out and in-going paths up to length K of all vertices:

$$X = \left[\underbrace{(\beta A^T)\mathbf{1} \cdots (\beta A^T)^K \mathbf{1}}_{\text{in-going paths}} \,|\, \underbrace{(\beta A)\mathbf{1} \cdots (\beta A)^K \mathbf{1}}_{\text{out-going paths}} \right], \tag{1}$$

where $\beta = \alpha/\lambda_{max}$ is a parameter that ensures convergence of the columns of (1) whenever $\alpha \in (0, 1)$ [5]. λ_{max} is the largest eigenvalue of A, while α is a scaling factor that weights the importance of longer paths, and hence, of the global (large α) or local (small α) structure of the network. For the current report, we choose $\alpha = 0.95$ and $K = 50$.

Two vertices are considered to be similar if they have similar path-profiles, i.e., vertices i and j are similar if their path vectors, represented by the i-th and j-th row vectors of X, are close according to any metric system between two vectors. Such metric is chosen to be the cosine distance: if $\mathbf{x}_i$ is the row vector for vertex i, the role-based similarity matrix Y is defined as

$$Y_{ij} = \frac{\mathbf{x}_i \mathbf{x}_j^T}{\|\mathbf{x}_i\| \|\mathbf{x}_i\|}. \tag{2}$$

Note that $Y_{ij} \in (0, 1)\ \forall i, j$, and $Y_{ij} \to 1$ if i and j share a very similar path-profile, while $Y_{ij} \to 0$ if they are dissimilar.

1.2 Dynamical embedding similarity

RBS takes into consideration the paths up to length K of the vertices, capturing the potential flow of information from a vertex to the rest of the network. Nevertheless, there is no explicit dynamics taking place in order to calculate (2). In this sense, embedding a network with a dynamical system can be a better description of the flows in it. Dynamical Embedding Similarity (DES) does precisely this: A network is embedded in a dynamical system, and the impulse response of the system gives us information about the flow of its vertices. For DES, we present the following description following[3].

Let $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ describe the dynamical response of a vertex i

$$\dot{\mathbf{s}}_i = \mathbf{f}(\mathbf{s}_i), \tag{3}$$

where $\mathbf{s}_i(t) \in \mathbb{R}^N$ is the impulse response of vertex i at time t. If $\mathbf{f}$ is a linear transformation, it can be represented as a matrix $\mathcal{L} \in \mathbb{R}^{N \times N}$, and the complete impulse response of the system is given by

$$S(t) = [\mathbf{s}_1(t) \cdots \mathbf{s}_N(t)] = \exp(\mathcal{L}\mathbf{t}). \tag{4}$$

$S(t)$ is the matrix containing the flow profile of the vertices acting under $\mathcal{L}$.

For this report we will take $\mathcal{L}$ to be the Laplacian matrix $L = D_{out} - A$, where $D_{out} = diag(d_{out})$ is the out-degree matrix, and $d_{i_{out}}$ is the out-degree of vertex i. This represents a random walker in a directed network, and, as stated in [3], it doesn't need to have a teleportation term. Short time responses of this dynamical system will give a local description of the flow of the vertices, while longer times will spread all through the network, giving a global description of the flows in the network.

Following the same fashion of section 1.1, the flow-response of each vertex can then be compared through any distance matrix, for which cosine distance will be chosen again. We construct the dynamical-embedding similarity matrix $\Psi(t)$[1] as

$$\Psi_{ij}(t) = \frac{\mathbf{s}_i(t)^T \mathbf{s}_j(t)}{\|\mathbf{s}_i(t)\| \|\mathbf{s}_j(t)\|}. \tag{5}$$

[1] In [3], they define similarity as the general standard bilinear product. We use the cosine distance in order to be able to compare DES with RBS.

Again, $\Psi_{ij}(t) \in (0,1)\ \forall i,j$, and $\Psi_{ij}(t) \to 1$ is i and j share a similar flow-profile, while $\Psi_{ij}(t) \to 0$ if they are dissimilar. We choose $t = 10$ for all of our tests to have a good balance of capturing local and global properties of the flows of the networks.

2 Similarity networks as a relaxed minimum spanning tree

Both Y and $\Psi(t)$ can represent the adjacency matrix of an undirected weighted network. Nevertheless, there is no edge sparsity in them, making modular clustering hard to handle. To overcome such density, we present a way to construct sparse similarity networks based on $\Psi(t)$ and Y by using the Relaxed Minimum Spanning Tree (RMST) algorithm, which incorporates both local and global features of the similarity matrices. We will follow the description of the RMST as in [4].

Take, without loss of generality, similarity matrix Y and define the "dissimilarity" matrix Z as $Z_{ij} = 1 - Y_{ij}$ and consider it as a (dense) weighted graph. Then, using Kruskal's algorithm [6], construct an unweighted Minimum Spanning Tree (MST). Each pair of vertices (i,j) is connected through a path $\{(i,k),(k,l),\cdots,(m,j)\}$ in the MST, for which we find the maximal weight in Z along it:

$$z_{ij}^{max} = \max\{Z_{ik}, Z_{kl}, \cdots, Z_{mj}\}. \tag{6}$$

Note that $Z_{ij} \to 0$ implies that i and j are similar, so if $z_{ij}^{max} \ll Z_{ij}$, then the MST is considered to be a good model to explain the similarity between i and j. If Z_{ij} is of the order of z_{ij}^{max}, it is not certaing that the MST path is a good model of the similarity between vertices i and j, so an edge between them is added if

$$Z_{ij} - \gamma\langle Z\rangle < z_{ij}^{max}, \tag{7}$$

where $\langle Z\rangle = 1/N^2 \sum_{ij} Z_{ij}$ is the average weight of the adjacency matrix Z,[2] and $\gamma \in (0,1)$ is a parameter that controls how dense will the RMST be, where we take $\gamma = 0.2$ for this report. The RMST consists of all the edges of the MST plus the ones that satisfy (7). The construction of the RMST for DES is analogous by constructing $Z_{ij} = 1 - \Psi_{ij}(t)$.

3 Obtaining community structure from similarity networks

To obtain community structure from the similarity networks we take motivation from the Louvain's method for modularity maximization developed in [7]. Nevertheless, in order to counteract the resolution limit of vanilla-modularity, we instead maximise the linear version of the stability of a Markov Process described by the normalised Laplacian of a graph, as constructed in [1].

The quality function developed in [1] is constructed by taking a Markov process $\mathcal{M}$ describing the motion of a random walker in an ergodic network, and then taking the stability of a partition $\mathcal{P}$ defined by

$$R_{\mathcal{M}}(t) = \sum_{C\in\mathcal{P}} \left(P(C,t) - P(C,\infty)\right), \tag{8}$$

where $P(C,t)$ is the probability of a walker initially at community C to be at community C at time t. As the network is ergodic, $P(C,\infty)$ is the probability of two independent walkers to be at C. Note that the RMST described in the previous section constructs undirected (unweighted) graphs, making them automatically ergodic.

The continuous-time Markov process for independent random walkers in a network is defined by

$$\dot{\mathbf{p}} = \left(D^{-1}A - I\right)\mathbf{p}, \tag{9}$$

where $p_i(\tau)$ is the density of walkers at vertex i at time τ, and $D = diag(k)$ is the diagonal degree-matrix, where k_i is the degree of vertex i.

The stationary solution of the process is $\mathbf{p}^* = D/2m$, with m the number of edges in the network, and, as (9) is a linear dynamical system, its general solution is given by $\mathbf{p}(\tau) = e^{\left(D^{-1}A-I\right)\tau}$. With these quantities, we can express (8) as

$$R_{\mathcal{M}}(\tau) = \sum_{C in \mathcal{P}} \sum_{i,j\in C} \left(\left[e^{\left(D^{-1}A-I\right)\tau} \right]_{ij} \frac{k_j}{2m} - \frac{k_i}{2m}\frac{k_j}{2m} \right), \tag{10}$$

[2] The decision of taking $\gamma\langle Z\rangle$ as the RMST-density term differs from [4]. Their decision of this term is related to the neighborhood of vertices i and j, while the term we choose is more related to the global behaviour of Z.

which gives us the Markov's stability of the process.

By linearising (10), we obtain

$$Q_{\mathcal{M}}(\tau) = (1-\tau) + \sum_{C\in\mathcal{P}} \sum_{i,j\in C} \left(\frac{A_{ij}}{2m}\tau - \frac{k_i k_j}{(2m)^2} \right), \tag{11}$$

which is a generalisation of modularity for any time resolution. There are two interesting things to notice from (11). First, $Q_{\mathcal{M}}(1)$ is exactly the vanilla-modularity as defined in [7]. And second, the generalised modularity doesn't suffer from the resolution limit of vanilla-modularity, as noted in [1]. For such reasons, (11) will be the quality function to optimise for this report, which is done by the standard procedure of Louvain's method.

In order to obtain the best optimal partition for a given network, we optimise (11) for a range of resolution values, and we look for the most robust set of partitions in such range. A partition will be robust if it is persistent over a range of Markov times and different optimisation iterations. A standard way to quantify this is through the normalised variation of information between partitions [8], which is defined as

$$\hat{V}(\mathcal{P}_1, \mathcal{P}_2) = \frac{2H(\mathcal{P}_2, \mathcal{P}_2) - H(\mathcal{P}_1) - H(\mathcal{P}_2)}{\log_2 N}, \tag{12}$$

where $H(\mathcal{P}_i)$ is the Shannon entropy of partition $\mathcal{P}_i$, and $H(\mathcal{P}_1, \mathcal{P}_2)$ is the Shannon joint entropy of the partitions. $\hat{V}(\mathcal{P}_1, \mathcal{P}_2) \in [0, 1]$ will be maximum when $\mathcal{P}_1$ and $\mathcal{P}_2$ are totally different, and it will be zero when the partitions are the same. For this report, we measure the similarity of partitions as done in [9], where we average $\hat{V}(\mathcal{P}(\tau), \mathcal{P}(\lambda\tau))$ over different iterations of Louvain's optimisation, with $\lambda = 20/19$.

4 Tests

In the following section, we test the optimal stability described in 3 of the similarity networks for both $\Psi(t)$ and Y of a directed scale-free graph and an undirected graph that represents a social network of dolphins. The decision-making for the resolution parameter in each case is presented in Appendix A, and the parameters for everything else are the ones mentioned in the text[3].

4.1 Scale-Free Network

The scale-free network $\mathcal{G}_{SF}$ is a directed network following a power-law degree distribution. It is constructed by the description in [10], using the standard parameters of python's *networkx* library [11]. It has and in-degree power-law exponent of 0.41 and an out-degree exponent of 0.05. The network has $N = 250$ vertices, $m = 441$ edges, and an average degree of 1.76.

Figure 1 presents the optimal partitions for the RBS similarity network (top) and the DES similarity network (bottom) of $\mathcal{G}_{SF}$, where their corresponding RMSTs are presented in the left, and $\mathcal{G}_{SF}$ is coloured in the right according to their corresponding RMSTs partitioning.

We observe interesting differences between how each of the similarity measures clusters $\mathcal{G}_{SF}$. RBS identifies three kind of roles in the networks, where the vertices on the purple community act as hubs, i.e., they have a big in-degree, and serve as vertices that transmit information to most of the network. Vertices in the red community are peripherical and have a low in-degree, which represent vertices that are casually referenced in the network but not big enough to serve as hubs. Vertices in the green community have almost exclusively zero in-degree and non-zero out-degree, meaning that they get information from the reds and the purples in order to interact in within the network.

DES also divides the network into three communities and, similar to RBS, the purple community represent the nodes that transmit information throughout the network, as they have zero out-degree and positive in-degree. Nevertheless, they don't discriminate how high their in-degree is, so the classification of hubs versus casual informers is lost. The other two communities are interpreted with a more structural scope, as the greens are taking information from the biggest hubs exclusively, while the reds are taking information from a secondary (but still big) hub. Notice that the green and the red communities don't interact with each other, meaning that their flows of information stay within themselves.

[3]These parameters are: $K = 50$ and $\alpha = 0.95$ for the RBS matrix, $t = 10$ for the DES matrix, and $\gamma = 0.2$ for the RMST.

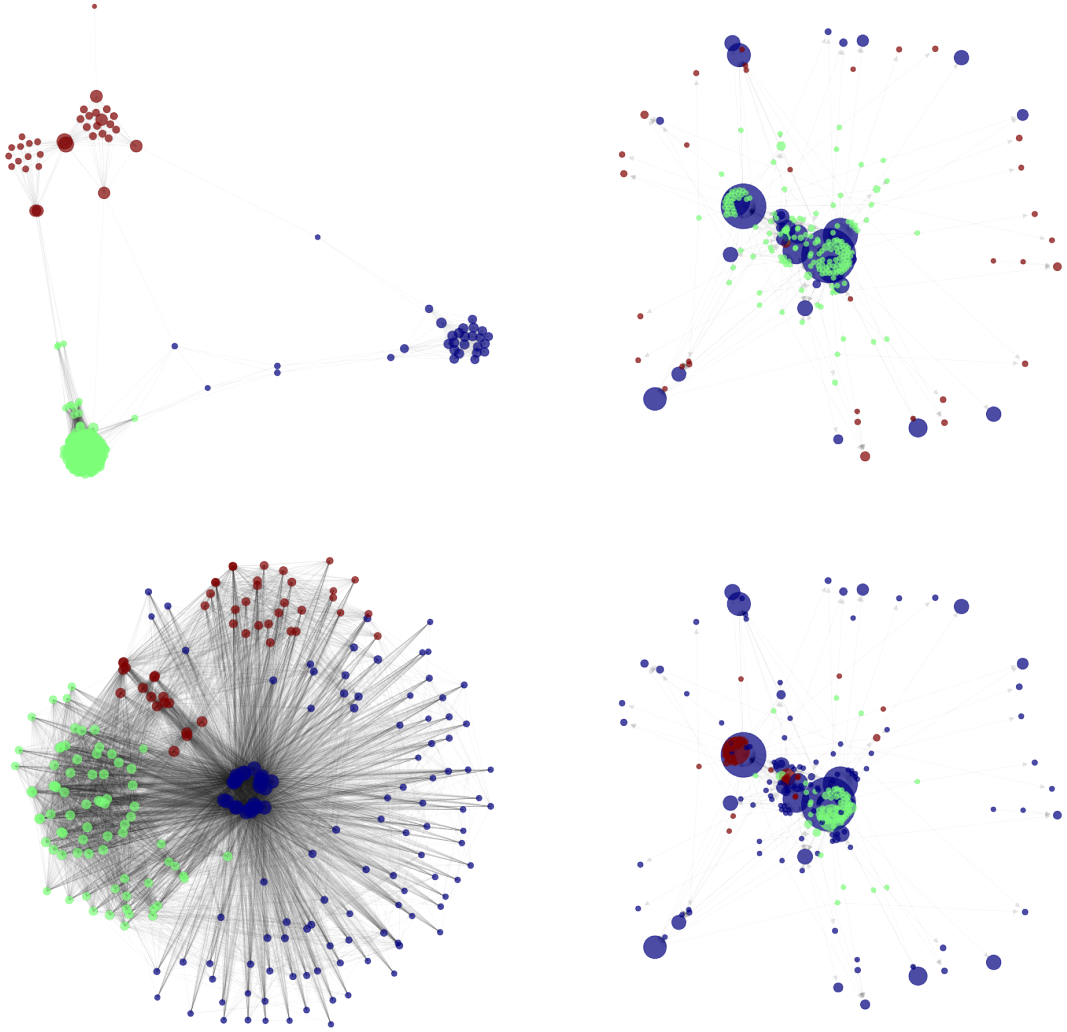

Figure 1: Optimal partitions of the RMST constructed from Y_{SF} for resolution parameter $\tau = 1.2$ (top) and $\Psi_{SF}(t)$ for resolution parameter $\tau = 1.5$. (bottom). Similarity networks (left) and $\mathcal{G}_{SF}$ (right) are coloured for the partition corresponding to their similarity measure. The size of the vertices is proportional to their PageRank.

4.2 Dolphin social network

For a second test, we take the social network of bottlenose dolphins $\mathcal{G}_D$ studied in [12]. It has $N = 62$ nodes, $m = 159$ edges, exhibits scale-free properties, and there is social structure within the dolphin society. Note that as the network is undirected, partitions obtained from optimising (11) are expected to exhibit more structural properties than in the previous example.

We take advantage of the undirectedness of the network to optimise (11) without applying any similarity measure to get the optimal partition of $\mathcal{G}_D$. This lets us compare not only the partitions for the RMSTs of the similarity measures, but also with the partition of the original network as well. The partition for $\mathcal{G}_D$ is presented in Figure 2 for its optimal Markov time, but it doesn't seem to show much information about the social structure of the network, as the partition is quite heterogeneous.

Figure 3 is equivalent to Figure 1 presented in the last example. Both RBS and DES similarity networks are very similar to the original dolphin graph, as can be seen in the figure. Nevertheless, their optimal partitions show very different results to that in Figure 2. The partition for the RBS similarity network clusters dolphins into two social communities (purples and reds), while a third community (green) is captured for dolphins that are in the periphery and interact with both the two main groups.

On the other hand, DES similarity network doesn't capture peripheral vertices as in RBS, but gets a more

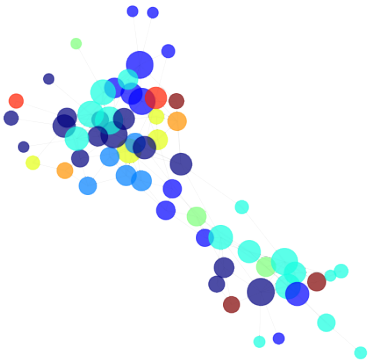

Figure 2: Optimal partition for $\mathcal{G}_D$ without any similarity measure for $\tau = 3$. The size of the vertices is proportional to their PageRank.

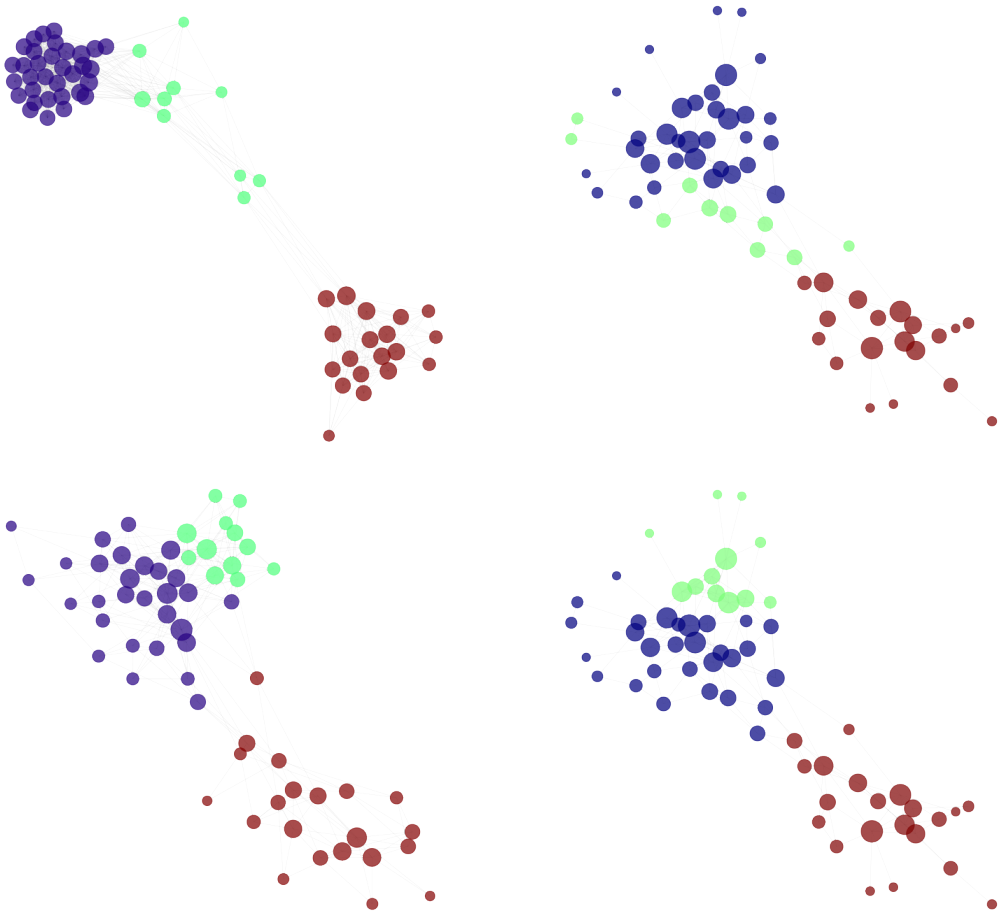

Figure 3: Optimal partitions of the RMST constructed from Y_D for resolution parameter $\tau = 2$ (top) and $\Psi_D(t)$ for resolution parameter $\tau = 1.3$. (bottom). Similarity networks (left) and $\mathcal{G}_{\mathcal{D}}$ (right) are coloured for the partition corresponding to their similarity measure. The size of the vertices is proportional to their PageRank.

structural classification of dolphins, where it seems to form communities of dolphins that are neighbours.

5 Discussion

We observe different properties of the networks depending on the similarity-measure used in the examples studied in this report. The role-based similarity matrix Y is a quantity considers all of the in-going and out-going paths of the network, so it is a global quantity up to the selected K path-length chosen. In this sense, it

captures the behaviour of vertices depending on the role they have in the network, independently if such vertices are neighbours in the original network or not. Dynamical-embedding similarity matrix $\Psi(t)$, on the other hand, is a dynamical quantity which is constructed from the Markov process of a random walker, so it will capture flows from the graph, and similar vertices will capture both global and local properties, at least for our choice of t.

Everything on this report was coded in python. The Minimum Spanning Tree is obtained with the `minimum_spanning_tree` method of the *networkx* library [11]. Community detection of section is found by using the `best_community` method of the *community* library [13]. The rest of the equations and algorithms used for this project can be found at my Github repository [14].

References

[1] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2008.

[2] Kathryn Cooper and Mauricio Barahona. Role-similarity based comparison of directed networks. *arXiv preprint arXiv:1103.5582*, 2011.

[3] Michael T Schaub, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. Multiscale dynamical embeddings of complex networks. *arXiv preprint arXiv:1804.03733*, 2018.

[4] Mariano Beguerisse-Díaz, Borislav Vangelov, and Mauricio Barahona. Finding role communities in directed networks using role-based similarity, markov stability and the relaxed minimum spanning tree. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 937–940. IEEE, 2013.

[5] Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

[6] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

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

[8] Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[9] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. Random walks, markov processes and the multiscale modular organization of complex networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90, 2014.

[10] Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Directed scale-free graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 132–139. Society for Industrial and Applied Mathematics, 2003.

[11] Official networkx source code repository. `https://github.com/networkx/networkx`.

[12] David Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(suppl_2):S186–S188, 2003.

[13] Louvain community detection. `https://github.com/taynaud/python-louvain`.

[14] Report source code repository. `https://github.com/blas-ko/Networks-Project`.

Appendices

A Decision of Markov time resolution for the similarity networks

The optimal partition for the similarity networks tested in this report, namely the scale-free and the dolphin networks, take a Markov time resolution based on the robustness of the variation of information in each case. Figure 4 and Figure 5 present the evolution of the number of communities over resolution range of $\tau \in [10^{-1}, 10^{2}]$ for the scale-free RBS RMST, scale-free DES RMST, dolphin RBS RSMT, and dolphin DES RSMT, respectively.

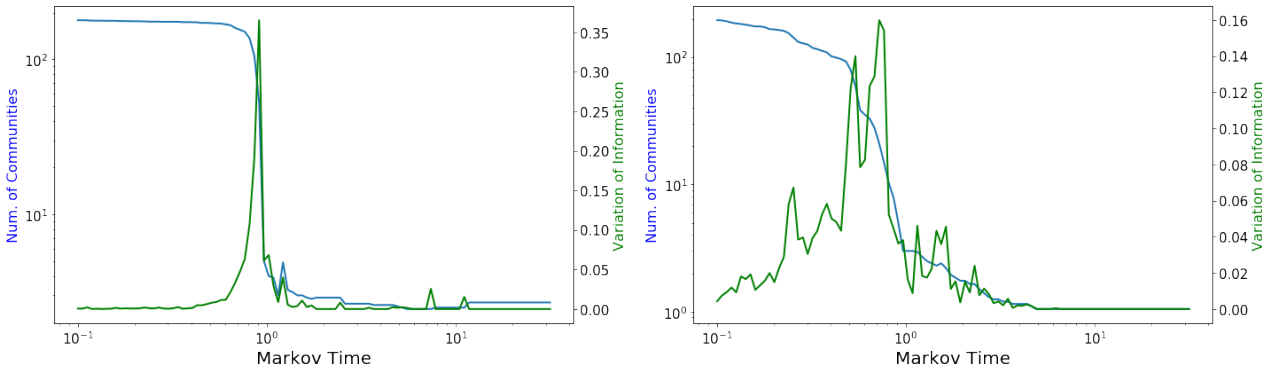

Figure 4: Number of communities in the left y-axis and $\hat{V}(\mathcal{P}(\tau), \mathcal{P}(\lambda\tau))$ for $\tau \in [10^{-1}, 10^{2}]$ for the RBS RSMT (left) and the DES RSMT (right) of the scale-free network. We choose the optimal Markov resolution time to be $\tau = 2$ for RBS and $\tau = 1.3$ for DES.

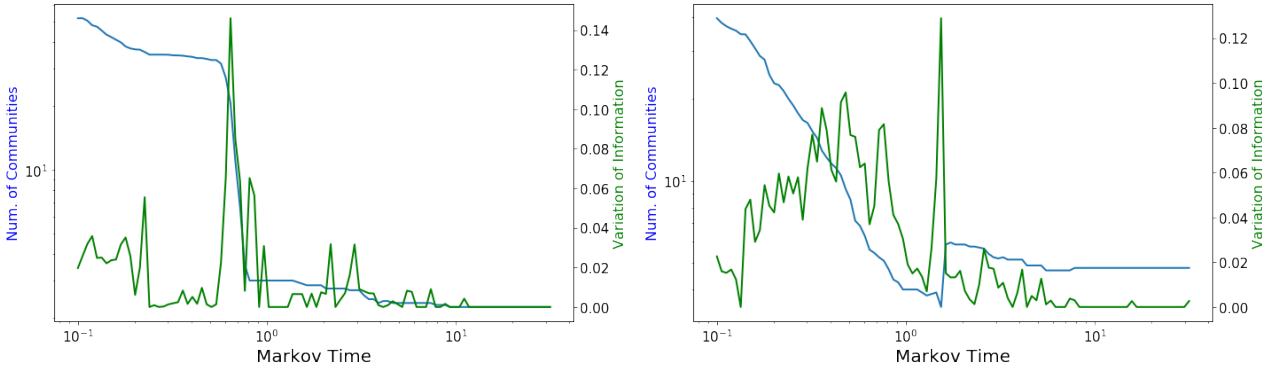

Figure 5: Number of communities in the left y-axis and $\hat{V}(\mathcal{P}(\tau), \mathcal{P}(\lambda\tau))$ for $\tau \in [10^{-1}, 10^{2}]$ for the RBS RSMT (left) and the DES RSMT (right) of the dolphin network. We choose the optimal Markov resolution time to be $\tau = 1.2$ for RBS and $\tau = 1.5$ for DES.