LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-645842

Which Network Similarity Measure Should You Choose: An Empirical Study

S. Soundarajan, T. Eliassi-Rad, B. Gallagher

November 7, 2013

Workshop on Information in Networks
New York, NY, United States
October 4, 2013 through October 5, 2013

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Which Network Similarity Measure Should You Choose: An Empirical Study

Sucheta Soundarajan
Rutgers University
s.soundarajan@cs.rutgers.edu

Tina Eliassi-Rad
Rutgers University
eliassi@cs.rutgers.edu

Brian Gallagher
Lawrence Livermore Lab
bgallagher@llnl.gov

Abstract

We consider the problem of determining how similar two networks, without known node-correspondences, are. This problem occurs frequently in real-world applications like transfer learning and change detection. Many network similarity measures exist, and it is unclear how one might select from amongst them. We provide the first empirical study on the relationships between different network similarity methods. Here, we propose (1) an approach for identifying groups of comparable network similarity methods and (2) an approach for computing the consensus among a given set of network similarity methods. We apply our approaches to seven real datasets and twenty network similarity methods. Our experiments demonstrate that (a) different network similarity methods are surprisingly well correlated, (b) some complex network similarity methods can be very closely approximated by much simpler methods, and (c) two network similarity methods–namely, random walk with restarts and NetSimile–provide similarity rankings that are closest to the consensus ranking.

Introduction

Assuming no knowledge of node-correspondences, how similar are two networks? We study a variety of network-similarity methods and address the following: (1) we identify whether the different similarity methods can be clustered into groups of methods that behave comparably and (2) we demonstrate how one can select a single consensus method from a group of similarity methods.

The study of network data covers diverse domains from social sciences to biology to information technology. While these different networks share important features, the extent of these similarities is not clear. A network similarity method is useful for applications such as detecting when the structure of an online financial network has changed, indicating possible fraud; or for determining when an algorithm developed on one network may be applied to a different network [2]. One network similarity method may compare two networks based on simple network features such as edge density, while another may examine more complex (and computationally burdensome) patterns such as communities.

We consider twenty network similarity methods on seven real datasets, applied to the task of *network-similarity ranking*, in which one is given a network G plus a set of other networks and must rank other networks in order of their similarity to network G. Our work yields several valuable results. First, we show that the various similarity methods, though seemingly different, produce correlated rankings. Second, we observe that some complex methods can be approximated by a much simpler method. For example, a method that compares random walks from two networks is well correlated with a method that simply measures density. Third, we describe how to select a single consensus from a set of rankings, and show that across all considered networks, two methods–namely, random walk with restarts and NetSimile [2]–are consistently closest to the consensus ranking.

Proposed Approach

Figure 1 presents an overview of our proposed approaches. Suppose that we are interested in the ranking behavior of r similarity methods. Given a network G_0 and a set of other networks $G_1, \ldots, G_k$, we employ the r methods to rank $G_1, \ldots, G_k$ by their similarity to G_0. This procedure produces r rankings of length k for G_0.

To determine ranking correlations, we find the *Kendall-Tau distance* between each pair of rankings. Given the rankings from a pair of methods, say $method_1$ and $method_2$, we calculate the difference between the probability that two randomly selected items from the rankings are in the same relative order versus the probability that those items are not in the same relative order. If two rankings are identical, their distance is 0. If one is the reverse of the other, their distance is 1. If the two rankings are uncorrelated, their distance is 0.5.

Next, we cluster the methods based on the pairwise Kendall-Tau distances. For this step, we choose *complete-linkage hierarchical clustering* because it produces a dendrogram with many small clusters, providing insight into which groups of methods are very closely correlated. The results of this clustering will indicate which groups of methods have comparable behavior. In particular, we are interested in learning whether any complex methods are associated with much simpler methods.

Finally, we use the *Kemeny-Young method* to combine the set of rankings into a single consensus ranking [3]. In this method, r rankings of k items are used to create a k-by-k preference matrix P, where P_{ij} is the number of

rankings that rank item i above item j. Next, each possible ranking R is assigned a score by summing elements P_{ij} for which R ranks i over j. The highest-scoring ranking is considered the consensus. Under the assumption that each ranking is a noisy estimate of a "true" ranking, the Kemeny-Young consensus is the maximum likelihood estimator for this true ranking. If some similarity method produces rankings that are very close to R, then one can simply use that method as a representative of the set of methods.

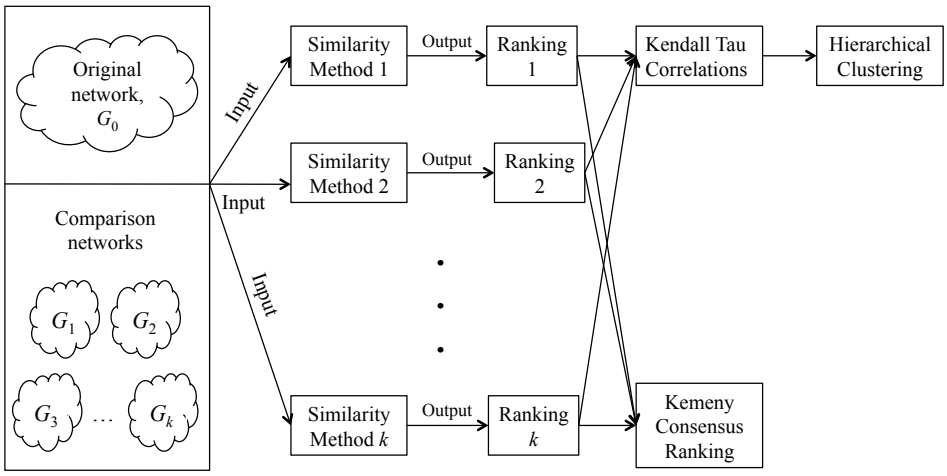

Figure 1: Our flowchart of how to compare various network similarity methods

Experiments

Our experiments include a set of twenty network similarity methods and seven real networks. Each of these twenty methods assigns a similarity score to a pair of networks. The similarity score is between 1 (perfect similarity) and 0 (perfect dissimilarity). Our datasets include two portions of the Facebook network, a portion of the Amazon co-purchasing network, the DBLP co-authorship network, two portions of the LiveJournal blogging network, and the Enron e-mail network. We refer the reader to [1] for details on these datasets. For each network, we create "baseline" networks for similarity comparisons by creating two synthetic networks. In the first baseline, we randomly delete 5% of the edges. In the second baseline, we randomly rewire 5% of the edges while preserving degree distribution.

For many of these similarity methods, we use the normalized Canberra distance to calculate the distance between two vectors. The Canberra distance between two numbers a and b is the ratio of $|a - b|$ to $|a + b|$. The normalized Canberra distance between two vectors $\vec{x}$ and $\vec{y}$ is the average of the Canberra distances of each pair of elements x_i and y_i. The normalized Canberra distance takes values between 0 and 1. Because we are interested in similarity, we subtract this value from 1 to obtain a similarity value between 0 and 1, where 1 indicates perfect similarity.

We divide the similarity methods into *community-level* and *network-level* methods. For the community-level methods, we randomly select 200 nodes in each network and find communities using breadth-first search (*BFS*), random walk without restart (*RW*), random walk with 15% chance of restart to the original node (*RWR*), and an alpha-beta community (*AB*). We represent each of the 200 communities with a feature vector describing its structural properties. We use features such as diameter, edge density, conductance, various centrality measures, *etc.* For the comprehensive set of features used, we refer the reader to [1]. Subsequently, we utilize these community-based feature vectors in two ways.

- *BFS, RW, RWR, AB*: We treat network similarity as a *classification* problem. To calculate the distance between networks G_0 and G_1, we label each community feature-vector with the name of the network from which it came. Then, we use an SVM classifier to perform cross-validation to determine whether the networks are separable. Each set of feature vectors is divided into a training set and a test set. A classifier is trained using the training set and then evaluated on the test set. For the G_0 (or G_1) feature vectors in the test set, some portion are correctly classified as coming from G_0 (or G_1), while the rest are incorrectly classified as coming from G_1 (or G_0). We create a length-2 vector containing the fractions classified as coming from G_0 and G_1. Subsequently, we take the Canberra distance between these vectors, and subtract this value from 1 to calculate the similarity between G_0 and G_1. If G_0 and G_1 are very similar, there will be a high rate of misclassification. In this case, we expect that each vector will be close to (0.5, 0.5) and the similarity will be

1. Conversely, if G_0 and G_1 are very dissimilar, we expect the classifier to have high accuracy. Specifically, we expect that the vector for G_0 to be close to (1.0, 0) and the vector for G_1 to be close to (0, 1.0). Therefore, the similarity will be close to 0.

- *BFS-Dist, RW-Dist, RWR-Dist, AB-Dist*: We create a complete bipartite graph in which the nodes in the first part of the bipartite graph correspond to community feature-vectors from network G_0 and the nodes in the second part of the bipartite graph correspond to community feature-vectors from network G_1. We weigh an edge between two nodes by the Canberra distance between the corresponding feature-vectors. Then, we find a *least cost matching*, divide the cost of this matching by 200 (i.e., the number of randomly selected nodes), and subtract the result from 1 to obtain the similarity value. If each feature-vector in network G_0 has an equal feature-vector in network G_1, then the cost of the matching is 0; and so the similarity is 1.

In the network-level methods, a network is assigned a feature vector. We calculate the similarity between two networks by subtracting from 1 the normalized Canberra distance between their vectors. The network-level methods are:

- *Eigenvalues*: A network is represented as a vector containing the ten largest eigenvalues of its adjacency matrix.
- *IM-In, IM-Known,* and *IM-In-&-Known*: We apply the InfoMap community detection method to a network. For each node u, where u is in community C, we calculate the fraction of neighbors of u that are in C and the fraction of nodes in C adjacent to u. We then average these values to form the length-one vectors of *IM-In* and *IM-Known*. *IM-In-&-Known* represents the network as a vector containing both values.
- *LBD*: The network is represented by a length-3 vector, which contains the network's triadic closure, the fraction of edges that share no common endpoints, and the extent to which a single vertex dominates the network [4].
- *Degree, Density,* and *Transitivity*: A network is represented by its average degree, density, or transitivity.
- *d-RW-Dist*: For values d = 10, 20, 50, and 100, we randomly select 100 nodes and perform a length-d random walk. Then, for each d, we measure the shortest path distance between the starting node and the ending node of its 100 walks. Finally, we define a length-20 vector by using the median and first four moments of the distributions for these shortest path distances of each d.
- *NetSimile, NetSimile-SVM, NetSimile-Dist*: NetSimile [2] represents each node by a length-7 vector containing its degree, clustering coefficient, average degree of its neighbors, average clustering coefficient of its neighbors, average clustering coefficient of its neighbors, number of edges outgoing from its egonet, and number of nodes adjacent to its egonet. A length-35 vector, containing the median and first four moments of distributions for these seven features over all nodes, represents the network. *NetSimile-SVM* and *NetSimile-Dist* are similar to the classifier-based (*BFS*, *RW*, *RWR*, and *AB*) and distance-based (*BFS-Dist*, *RW-Dist*, *RWR-Dist*, *AB-Dist*) methods described above, but instead of classifying networks through their community feature-vectors, we use the length-7 node-based feature-vectors.

Results

In nearly every case, each similarity method ranked the two baseline versions of each network as most similar to that network. The average Kendall-Tau distance between rankings, over all networks and all metrics, is 0.28 with a standard deviation of 0.14. Recall that a distance of 0 indicates perfect similarity. The different methods are thus usually correlated with one another even though they have different objective functions. Methods *RW* and *RWR* have an average distance over all networks of 0.09. This low distance (or alternatively, high correlation) is expected because the two methods are very similar. In other cases, the results are surprising. *NetSimile* and *RWR* have an average distance of 0.12, despite being very different. This suggests that although some methods are complex, similar behavior can be achieved by a simpler method.

In general, methods that are highly correlated on one dataset are also highly correlated on other datasets. To measure this, for each network, we create a length-190 vector containing the Kendall-Tau distances between each pair of similarity methods. This produces seven vectors, one corresponding to each dataset. We then calculate the Canberra distances between these vectors. If the distance between the vectors for G_0 and G_1 is low, this indicates that the methods that are well correlated on network G0 are also well correlated on network G_1, and the methods that are poorly correlated on network G_0 are also poorly correlated on G_1. Across all pairs of networks, the average Canberra distance between these vectors is 0.097 with a standard deviation of 0.026. This low distance signifies that correlations between the methods are similar across different networks.

In order to quantify the relationships between methods, we cluster the methods using complete-linkage hierarchical clustering based on pairwise Kendall-Tau distance. Here, we are interested in learning whether groups

of complex methods are associated with simpler, more intuitive methods, such as *Density*. For each network, we perform the clustering 1000 times and select the most common dendrogram. We observe certain clusters across many of these dendrograms. Table 1 lists clusters observed in four or more networks out of the seven considered. We see that there are groups of methods that behave comparably. In some cases, these groups contain a mix of both complex as well as simple methods. For example, *RW-Dist*, *RWR-Dist*, and *BFS-Dist* behave very much like the simpler *Density* method. This suggests that for future network similarity tasks, one should use the computationally more efficient *Density* method as a replacement for these computationally intensive community-based methods.

Cluster	**Networks**
IM-In-&-Known, IM-Known	All 7 networks: Amazon, DBLP, Enron, Facebook1, Facebook2, LiveJournal1, LiveJournal2
RW-Dist, RWR-Dist	All 7 networks: Amazon, DBLP, Enron, Facebook1, Facebook2, LiveJournal1, LiveJournal2
RW, RWR, BFS, NetSimile-SVM	5 out of 7 networks: DBLP, Enron, Facebook2, LiveJournal1, LiveJournal2
LBD, Transitivity	5 out of 7 networks: Amazon, DBLP, Enron, LiveJournal1, LiveJournal2
NetSimile-Dist, IM-In	4 out of 7 networks: Amazon, Enron, LiveJournal1, LiveJournal2
RW-Dist, RWR-Dist, BFS-Dist, Density	4 out of 7 networks: Amazon, Enron, LiveJournal1, LiveJournal2

Table 1: Clusters observed in the dendrograms of at least four networks in 1000 runs per network.

After generating 1000 dendrograms, for each network, we generate a summary dendrogram describing clusters across all networks. To produce this clustering, we use the 1000 dendrograms created for each of the seven networks. For each of these 7000 dendrograms and for each pair of similarity methods $method_i$ and $method_j$, we calculate the path distance from $method_i$ to $method_j$ in the dendrogram. If $method_i$ and $method_j$ are clustered together early in the dendrogram, this distance is short. We next create a 20-by-20 summary matrix M in which the rows and columns correspond to the various similarity methods, where M_{ij} is defined as the sum of the path distances from $method_i$ to $method_j$ over all 7000 dendrograms. We then perform complete-linkage hierarchical clustering on the summary matrix. If $method_i$ and $method_j$ are consistently close to each other in the 7000 dendrograms, they will be clustered together in the summary dendrogram. Figure 2 contains the summary dendrogram resulting from this process. Some clusters are expected. For example, *RW-Dist*, *RWR-Dist*, and *BFS-Dist* are clustered together. More surprisingly, the complex methods *NetSimile-Dist* and *NetSimile* are clustered with the *Degree* ranking.

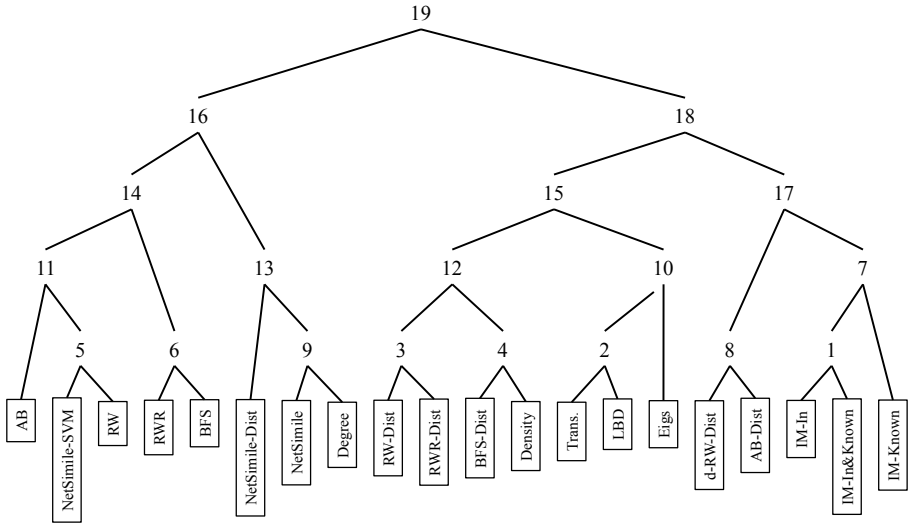

Figure 2: Summary dendrogram generated through complete-linkage hierarchical clustering. Observe that some complex methods (such as *RW-Dist*, *RWR-Dist*, and *BFS-Dist*) are clustered with much simpler methods (such as *Density*). Numbers inside inner nodes indicate the order in which clusters were joined.

Lastly, we apply the Kemeny-Young method to obtain a single consensus ranking. Table 2 lists the five similarity methods that are closest to this consensus for each network, as measured by the Kendall-Tau distance. Figure 3 contains a heatmap depicting the Kendall-Tau distance between each similarity method and the Kemeny-

Young consensus on each network. We observe some variations across experiments for different networks, but *NetSimile* (or one of its variations) and *RWR* appear in the top five positions for each network. Moreover, *RWR* has an average Kendall-Tau distance of 0.06 from the consensus, averaged over all networks. However, *RWR* has an average Kendall-Tau distance of 0.21 from the other similarity methods. This suggests that it is consistently close to the consensus (i.e., median) ranking, but not because it is simply close to the other rankings in general. A user interested in selecting a single method for network-similarity ranking should simply select NetSimile or RWR.

Amazon	DBLP	Enron	Facebook1	Facebook2	LiveJournal1	LiveJournal2
NetSimile-SVM	AB-Dist	NetSimile-SVM	NetSimile	NetSimile-Dist	RWR	RWR
RWR	RWR	BFS-Dist	NetSimile-Dist	NetSimile	Eigenvalues	Eigenvalues
IM-In-&-Known	Degree	RWR	RWR	RWR	BFS	BFS
RWR-Dist	NetSimile-SVM	BFS	BFS	RW	AB	RW
IM-Known	BFS	RW	Transitivity	Degree	NetSimile	NetSimile-SVM

Table 2: The five similarity methods closest to the Kemeny-Young consensus.
NetSimile (or one of its variations) appears at least once in every column, and RWR appears in every column.

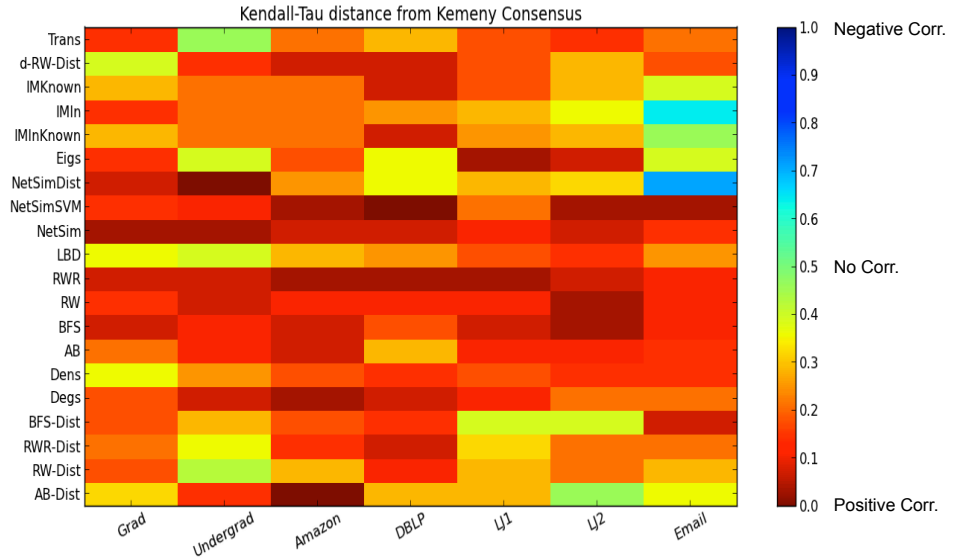

Figure 3: Kendall-Tau distance between each similarity method and the Kemeny-Young consensus on each network. Some methods, such as *NetSimile* and *RWR*, are often very close to the consensus.

Conclusion

We described two approaches for empirically analyzing a set of twenty network-similarity methods. Our application was ranking a set of networks based on their similarity to a given network. We calculated Kendall-Tau distances between the rankings produced by different similarity methods, and used these values to group methods that behave comparably. Moreover, we demonstrated how to use the Kemeny-Young method to select a single consensus ranking. Our analysis revealed that (1) various similarity methods have smaller than expected differences, indicating that "different" methods behave comparably when it comes to ranking applications; (2) simple similarity methods can closely approximate more complex ones; and (3) rankings produced by NetSimile and RWR are close to the Kemeny-Young consensus ranking.

References

[1] Bruno D. Abrahao, Sucheta Soundarajan, John E. Hopcroft, Robert D. Kleinberg: On the separability of structural classes of communities. KDD 2012.

[2] Michele Berlingerio, Danai Koutra, Tina Eliassi-Rad, Christos Faloutsos: NetSimile: A Scalable Approach to Size-Independent Network Similarity. ASONAM 2013.

[3] John Kemeny: Mathematics without numbers. Daedalus 88: 577–591 (1959).

[4] Owen Macindoe, Whitman Richards: Comparing networks using their fine structure. IJSCCPS 1(1): 79-97 (2011).

Prepared by LLNL under Contract DE-AC52-07NA27344.