A Guide to Selecting a Network Similarity Method

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

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

Brian Gallagher
Lawrence Livermore Laboratory
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 such as transfer learning and change detection. Many network-similarity methods exist; and it is unclear how one should select from amongst them. We provide the first empirical study on the relationships between different network-similarity methods. Specifically, we present (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 compare and contrast twenty network-similarity methods by applying our approaches to a variety of real datasets spanning multiple domains. Our experiments demonstrate that (1) different network-similarity methods are surprisingly well correlated, (2) some complex network-similarity methods can be closely approximated by a much simpler method, and (3) a few network-similarity methods produce rankings that are very close to the consensus ranking.

1 Introduction

How similar are two networks assuming we have no known node-correspondences between them? We study a variety of network-similarity methods in cross-sectional and longitudinal settings, and address the following questions: (1) How correlated are different network-similarity methods to each other? (2) How can one automatically find groups of methods that behave comparably? (3) How can one select a single consensus method from a group of network-similarity methods?

The study of networked data covers diverse domains from social sciences to physics to biology to information technology. While different networks can 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 a network has changed (possibly indicating fraud in an online financial network); or for determining when a classifier trained on one network may be applied to a different network [8]. A network-similarity method might compare two networks based on simple features such as edge density, or may examine more complex (and computationally burdensome) patterns such as communities.

We consider twenty network-similarity methods applied to a variety of networks from diverse domains. We consider the task of network-similarity ranking, in which one is given a reference network G as well as a set of other networks, and must rank those other networks in order of their similarity to G. Within the context of the ranking application, we present an approach to identify correlations between similarity methods, cluster methods, and select a consensus (or median) ranking from a group of rankings.

Our experiments are two-pronged. First, we apply our approaches to a set of cross-sectional datasets, demonstrating several valuable results. (1) We show that the various similarity methods, although seemingly different, produce well-correlated rankings. (2) 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. (3) We show that two methods – namely, NetSimile [6] and Random Walk with Restarts – are consistently close to the consensus. Second, we apply our approaches to a set of longitudinal datasets. We consider three datasets, each containing multiple networks aggregated on a daily or monthly basis. Our analysis of these networks reveals complexities in measuring network similarity over time. We discuss topics such as selecting an appropriate time granularity for longitudinal data. When an appropriate time granularity is used, we again observe high correlations between different network similarity methods.

Contributions. The major contributions of our paper can be summarized as follows:

- We categorize network-similarity methods and introduce novel approaches for comparing them.
- We conduct the first large-scale empirical study of network-similarity methods. Our experiments demonstrate the following: (1) Different methods are surprisingly well correlated. (2) Some complex methods are closely approximated by much simpler methods. (3) A few methods produce similarity

rankings that are very close to the consensus ranking.

- We provide practical guidance in the selection of an appropriate network similarity method.

The rest of the paper is organized as follows. We provide some background next. We then present our categorization of network-similarity methods in Section 3 and our comparison approaches in Section 4. These are followed by our experiments and discussion in Sections 5 and 6, respectively. We conclude the paper in Section 7.

2 Background

Quantifying the difference between two networks is critical in applications such as transfer learning and change detection, and a variety of network similarity methods have been proposed for this task (see Section 3 for details). The roots of this problem can be traced to the problem of determining graph isomorphism, for which no known polynomial-time algorithm exists. The network similarity task is much more general. For example, two networks can be similar without being isomorphic. There are many network similarity algorithms that require known node-correspondences. Examples include DeltaCon [11] and most edit-distance based methods.

3 Network Similarity Methods

We categorize a network-similarity method based on two criteria. First, at what level of the network does it operate? Second, what type of comparison does it use? For the first criterion, we define three levels: *micro*, *mezzo*, and *macro*. As their names suggest, at the micro-level a method extracts features at the node- or egonet-level;[1] at the mezzo-level it extracts features from communities; and at the macro-level it extracts features from the global/network level. For the second criterion, we have three types: *vector-based*, *classifier-based*, and *matching-based*. We describe each of these types below.

Vector-based methods assign feature vectors F_1 and F_2 to each network G_1 and G_2, respectively. They define the similarity between G_1 and G_2 as $1 - Canberra(F_1, F_2)$ [12].[2]

Classifier-based methods first identify a fixed number of structures within each network (such as random walks, communities, or node neighborhoods). For each of these structures, they calculate a feature vector describing its structural properties (e.g., the number of edges within a node neighborhood); and label these feature vectors with the name of their respective network. Then, using cross-validation, they determine whether an SVM can accurately distinguish between the feature vectors from network G_1 and the feature vectors from network G_2. In each round of cross-validation, the test set contains feature vectors from G_1 and G_2; and so for each of G_1 and G_2, they create a length-2 feature vector (respectively, F_1 and F_2) describing the fraction of feature vectors from that network that were classified as belonging to G_1 and the fraction that were classified as belonging to G_2. They define the similarity between G_1 and G_2 as $1 - Canberra(F_1, F_2)$. If G_1 and G_2 have very similar local structure, then we expect that SVM will not be able to distinguish between the two classes of feature vectors, and F_1 and F_2 will be very similar. The distance between F_1 and F_2 will be very low, and so the similarity will be high. Conversely, if G_1 and G_2 have very different structures, then the SVM will have high classification accuracy, and a low similarity score.

Matching-based methods use the same structures and feature vectors obtained in the classifier-based methods. However, instead of using a classifier to distinguish between the two classes, they match feature vectors from G_1 with similar feature vectors from G_2; and calculate the cost of this matching. Specifically, they create a complete bipartite graph in which nodes in the first part correspond to feature vectors from G_1 and nodes in the second part correspond to feature vectors from G_2. The weight of an edge in this bipartite graph is the Canberra distance between the corresponding feature vectors. They then find a least-cost matching on this bipartite graph, and the similarity is 1 minus the average cost of edges in the matching. If every feature vector in G_1 has an equal feature vector in G_2, the cost of the matching is 0, and so the similarity is 1. If the feature vectors from G_1 and G_2 are very different, the matching is more costly and the similarity is low.

Table 1 categorizes our network-similarity methods based on the aforementioned two criteria. We briefly define each of these twenty method below. Because macro-level methods consider the entire network at once, rather than local sub-structures, it is not possible for such methods to be classifier- or matching-based.

- NetSimile [6]: It represents a network by a vector describing its local structural features. NetSimile begins by calculating seven features for each node: the degree of the node, the clustering coefficient of the node, average degree of the node's neighbors, average clustering coefficient of the node's neighbors, the number of edges in the node's egonet, the

[1] Egonet is the 1-hop induced subgraph around the node.

[2] $Canberra(U, V) = \sum_{i=1}^{n} \frac{|U_i - V_i|}{|U_i| + |V_i|}$, where n is the number of dimensions in U and V. [12]

	Micro-level	Mezzo-level	Macro-level
Vector-based	NetSimile	Random Walk Distances, InfoMap-In, InfoMap-Known, InfoMap-In&Known	Degree, Density, Transitivity, Eigenvalues, LBD
Classifier-based	NetSimileSVM	AB, BFS, RW, RWR	–
Matching-based	NetSimile-Dist	AB-Dist, BFS-Dist, RW-Dist, RWR-Dist	–

Table 1: The twenty network-similarity methods considered in this paper categorized by two criteria. First, at what level of the network does the method operate? Second, what type of comparison does the method use? All of these methods assume no known node-correspondences.

number of edges outgoing from the egonet, and the number of nodes adjacent to the egonet. For each of these seven features, NetSimile has a distribution over all nodes; and calculates each feature's median and first four moments of distribution. These five values over seven features give a length-35 vector. Then, two networks can be compared by calculating the distance between their feature vectors.

- NetSimileSVM: For each network, we identify 300 node neighborhoods and calculate the 7 statistics used in the NetSimile method. We then apply the classification-based process described earlier.

- NetSimile-Dist: We use the same node neighborhoods and feature vectors obtained by NetSimileSVM, and apply the matching procedure.

- Random Walk Distances (d-RW-Dist for short): For each network, for values $d = 10$, 20, 50, and 100, perform 100 random walks of length d. Each of these walks begins on a randomly selected node n and terminates on some other node m. Calculate the shortest path distance between n and m in the network. For each value of d, aggregate these distances over all 100 random walks. Calculate the median and first four moments of distribution for this set of values. Over all four values of d, this produces a length-20 feature vector.

- InfoMap-In (IMIn), InfoMap-Known (IMKnown), InfoMap-In&Known (IMIn&Known): Apply the Infomap community detection algorithm to the network [16]. For each node n, identify which community C that node n is in. IMIn creates a length-1 feature vector containing the fraction of each node's neighbors that are in the same community as the node, averaged over all nodes. IMKnown creates a length-1 feature vector containing the fraction of nodes in C that are adjacent to n, averaged over all nodes. IMIn&Known creates a length-2 feature vector containing both of these values.

- AB, BFS, RW, RWR: For each network, we identify 300 communities. These communities are produced, respectively, by the $\alpha-\beta$ swap algorithm [7], breadth-first-search, random walk without restart, and random walk with 15% chance of restart to the original node. For each of these communities, we calculate a length-36 feature vector including statistics such as conductance, diameter, density, and so on. The full feature vector is described in [4]. We then apply the classification process described earlier to calculate network similarity.

- AB-Dist, BFS-Dist, RW-Dist, RWR-Dist: We use the same communities and feature vectors as identified by methods AB, BFS, RW, and RWR. We then perform the matching procedure described above.

- Density, Degree, Transitivity: For each network, we create a length-1 feature vector containing the density, average degree, or transitivity of that network. We then use the vector-based procedure describe earlier to compute network similarity.

- Eigenvalues (Eigs): For each network, we calculate the k largest eigenvalues.[3] This defines a length-k feature vector. We then use the vector-based procedure describe earlier to compute network similarity.

- LBD (short for Leadership-Bonding-Diversity): LBD calculates three features from each network [15]. The leadership statistic measures how much the connectivity of the network is dominated by one vertex. It is calculated by identifying the highest degree node in the network, and then averages the difference between that degree and the degree of every other node in the network. The bonding statistic is simply the transitivity of the network, and is measured by dividing the number of closed triads (triangles) in the network by the total number of open or closed triads in the network. Diversity calculates the number of disjoint dipoles,

[3] We used $k = 10$ in our experiments.

and normalizes this value to a similar range as the leadership and bonding statistics. Using these three values, networks are plotted on a simplex, and one can then visually determine which networks are similar. When using the LBD method, we represent each network by its length-3 vector, and take the difference between two vectors to calculate similarity.

4 Comparing Network Similarity Methods

We are interested in analyzing the relationships between different network-similarity methods. In particular, we (1) determine the correlations between different methods, (2) locate clusters of methods that behave similarly, and (3) identify methods that produce results that summarize the collection of results.

Figure 1 contains an overview of our process. In particular, we approach this problem from the application of network-similarity ranking. In this application, we are given some reference network G_r and a collection of comparison networks H_1, H_2, $\cdots$, H_k. Using a network-similarity method, we calculate the similarity between G_r and each H_i, and then rank the comparison networks in order of their similarity to the reference network G_r. By considering the problem from the perspective of ranking rather than considering raw similarity scores, we are able to compare similarity methods that may generate similarity scores across very different ranges. Given a reference network, a collection of comparison networks, and m network similarity methods, we produce m rankings of the comparison networks. We then compare the m rankings to one another in order to determine similarity between the various methods.

To determine ranking correlations, we find the Kendall-Tau distance between each pair of rankings. Given the rankings from a pair of methods m_1 and m_2, the Kendall-Tau distance between the rankings is the probability that two randomly selected items from the rankings are in different relative orders in the two rankings. A distance of 0 indicates perfect correlation, a distance of 0.5 indicates no correlation, and a distance of 1 indicates an inverse correlation.

To find methods that have comparable behavior, we cluster the methods based on the pairwise Kendall-Tau distances. For this step, we use complete-linkage hierarchical clustering because it tends to produce a dendrogram with many small clusters, which in turn provides insight into which groups of methods are very closely correlated. For each reference network, we perform the complete-linkage hierarchical clustering l times.[4] We then select the most common (i.e., representative) dendrogram by (1) considering each dendrogram as a tree without information about clustering order and (2) picking the tree that occurs most frequently out of these l runs as the representative dendrogram. The results of this clustering 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.

To obtain the consensus ranking, we use the Kemeny-Young method to combine the set of rankings into a single consensus ranking [9]. In this method, m 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 all 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 of this true ranking. If some similarity method consistently produces rankings that are very close to R, then one can use this method as a representative (i.e., consensus) of the set of methods.

5 Experiments

In this section, we describe our datasets, methodology, and experiments on cross-sectional and longitudinal data.

5.1 Datasets We use a variety of network datasets spanning multiple domains. Our experiments are performed on cross-sectional data representing the state of a network at one moment in time; and on longitudinal datasets, each containing multiple copies of a network that changes over time. Table 2 presents statistics for all of our datasets.

Our cross-sectional datasets are as follows. **Grad** and **Undergrad**: portions of the Facebook network corresponding to graduate and undergraduate students at Rice University [14]. **DBLP**: a computer science co-authorship network [1]. **LJ1** and **LJ2**: portions of the LiveJournal blogging network [5]. **Enron**: the Enron e-mail network [10]. **Amazon**: a portion of the product co-purchasing network from Amazon.com [13].

Our longitudinal datasets are as follows. **Twitter Replies**: a collection of 30 networks representing replies on Twitter, aggregated daily over the period of 30 days in June 2009 [2]. **Twitter Retweets**: a collection of 5 networks representing retweets on Twitter, aggregated monthly from May through September of 2009 [2]. **Yahoo! IM**: a collection of 28 networks representing conversations on the Yahoo Instant Messaging platform, aggregated daily during April 2008 [3].

[4] In our experiments, we used $l = 1000$.

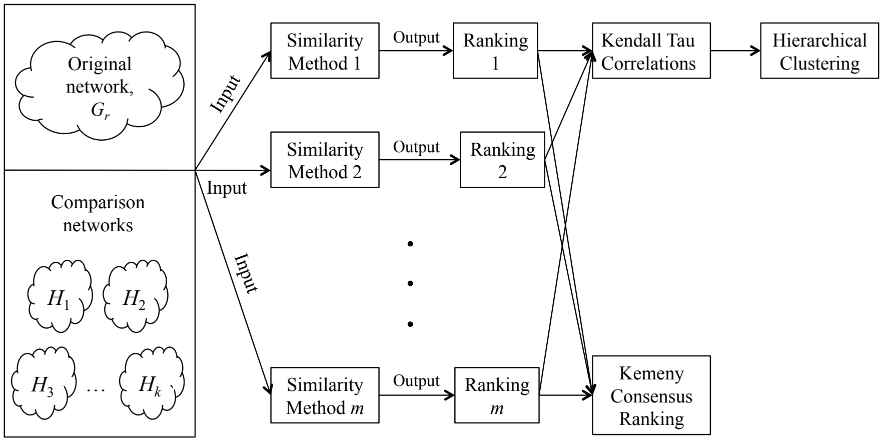

Figure 1: Flowchart of our two approaches. In our first approach, we calculate correlations between different methods and cluster the methods. In our second approach, we identify the consensus.

Network Name	**# of Nodes**	**# of Edges**	**Avg. Deg.**	**Max. Deg.**	**Edge Density**	**Network Transitivity**	**# of Comps**	**Frac. Nodes in LCC**
Grad	500	3000	13	48	0.03	0.43	2	0.996
Undergrad	1220	43K	71	287	0.06	0.24	1	1.0
Amazon	270K	741K	6	324	0.00002	0.21	4K	0.915
DBLP	740K	2.5M	7	705	0.00001	0.23	36K	0.851
LJ1	500K	11M	43	16,365	0.0001	0.04	1	1.0
LJ2	500K	11M	43	12,796	0.0001	0.08	1	1.0
Email	37K	184K	10	1,383	0.0003	0.09	1K	0.918
Twitter Replies	10K–27K	7K–21K	1.3–1.4	26–147	4.7×10^{-5} – 13×10^{-5}	0.0–0.001	4K–9K	0.007–0.05
Twitter Retweets	25K–120K	28K–165K	2.1–2.8	300–1300	2.3×10^{-5} – 8.5×10^{-5}	0.02–0.03	3K–6K	0.63–0.84
Yahoo! IM	28K–100K	35K–180K	2.5–3.6	66–123	3.6×10^{-5} – 8.5×10^{-5}	0.08–0.20	600–3K	0.48–0.85

Table 2: Statistics for Network Datasets. For longitudinal datasets, each dataset contains multiple networks, and so a range of values are presented for each value. Observe the large variations in statistics across different datasets.

5.2 Methodology At the heart of our approach is a set of 20 network-similarity methods described in Section 3. Each of these methods compares two networks and outputs a similarity score between 0 and 1. A score of 0 indicates that the networks are completely dissimilar and a score of 1 indicates that the networks are completely similar.

5.3 Experiments on Cross-Sectional Data We begin by applying our comparison approaches to the 7 cross-sectional datasets and 20 network similarity methods described earlier. We consider each of the 7 networks individually as a reference network. For each reference network, we produce two baseline networks by deleting a random 5% of edges and by rewiring a random 5% of edges in such a way as to preserve degree distribution. We then use the 20 methods to rank the other 8 networks (including the 2 baseline networks) relative to the reference network. We calculate the Kendall-Tau distances between each pair of these 20 rankings. 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 correlation. Figure 2 contains the Kendall-Tau distances between the different methods for the case when DBLP was used as a reference graph. Surprisingly, the different methods are usually correlated with one another even though they have different objective functions. Methods RW and RWR have an average distance across all networks of 0.09.

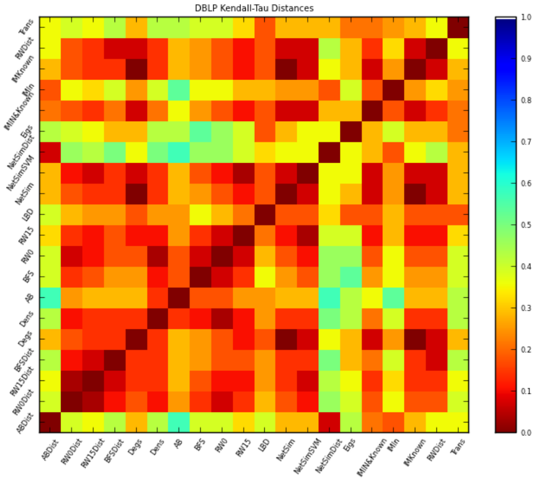

Figure 2: Heatmap showing Kendall-Tau distances between network similarity rankings when network DBLP was used as the reference network. Distances are generally low, indicating high correlations between rankings.

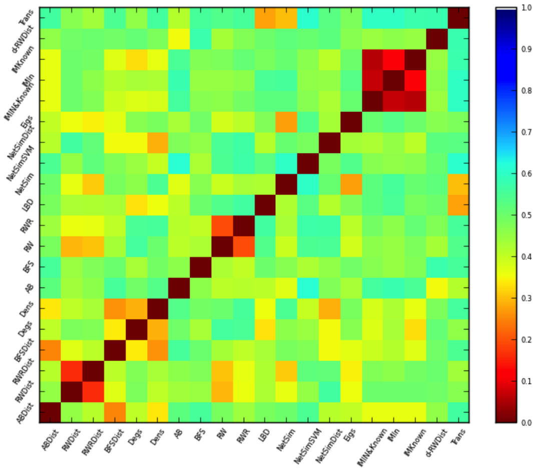

Figure 3: Kendall-Tau distances between rankings on network Twitter Replies. Observe that distances are very close to 0.5, indicating a lack of correlation.

This low distance (or alternatively, high correlation) is expected because the two methods are very similar; but in other cases, the results are more surprising. For instance, NetSimile and RWR have an average Kendall-Tau distance of 0.12, despite having very different objective functions.

Next, we cluster the methods using complete-linkage hierarchical clustering on the pairwise Kendall-Tau distances. Here, we are interested in learning whether groups of complex methods are associated with simpler, more intuitive methods. For each reference network, we perform the clustering 1000 times and select the most common dendrogram. We observe certain clusters across many of these dendrograms. Table 3 lists clusters observed in four or more clusters out of the seven considered. Some clusters 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 much simpler Density method. This suggests that for frequent network similarity tasks, one could use the computationally more efficient Density method as a replacement for these computationally intensive community-based methods.

Lastly, we apply the Kemeny-Young method to obtain a single consensus ranking. Table 4 lists the five similarity methods that are closest to this consensus for each network, as measured by Kendall-Tau distance. NetSimile (or one of its variations) and RWR appear in the top five positions for each network. 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 representative method for network similarity ranking should thus simply select NetSimile or RWR.

5.4 Experiments on Longitudinal Data In these experiments, each dataset contains multiple networks aggregated monthly (Twitter Retweets) or daily (Yahoo IM and Twitter Replies). Previous work [6] identified an anomalous network in each of these datasets, which upon examination, proved to correspond to important real-world events such as the Iranian presidential elections exhibiting online such as in the Twitter Retweets graph. For each of these datasets, we use these anomalous networks as the reference networks. The choice of the reference network is not a key element of our study. Similar to before, we produce baseline versions of the reference networks by deleting and rewiring edges.

We next calculate the Kendall-Tau distances between the rankings produced by the different methods. Figures 3 and 4 show correlations on Twitter Replies and Twitter Retweets (Yahoo! IM behaves similarly to Twitter Retweets). The distances between methods on these datasets are higher than the distances seen on the cross-sectional datasets (so the correlations are lower). The distances on the Twitter-Retweet datasets are still low, indicating positive correlations, and the distances on Yahoo! IM are also generally below 0.5. On the Twitter Replies dataset, however, the distances are typically all very close to 0.5, indicating no correlation.

To answer the question of why correlations on

Cluster	Networks
IMIn&Known, IM Known	All networks: Grad, Undergrad, Amazon, DBLP, LJ1, LJ2, Email
RW-Dist, RWR-Dist	All networks: Grad, Undergrad, Amazon, DBLP, LJ1, LJ2, Email
RW, RWR, BFS, NetSimileSVM	5 networks: Undergrad, DBLP, LJ1, LJ2, Email
LBD, Transitivity	DBLP, Amazon, LJ1, LJ2, Email
NetSimile-Dist, IMIn	4 networks: Amazon, LJ1, LJ2, Email
RW-Dist, RWR-Dist, BFS-Dist, Density	4 networks: Amazon, LJ1, LJ2, Email

Table 3: Clusters that appear in the most common dendrogram for at least four out of the seven reference networks. Interestingly, complex methods often appear in clusters with simpler methods.

Grad	Undergrad	Amazon	DBLP	LJ1	LJ2	Email
NetSimile	NetSimile-Dist	NetSimileSVM	AB-Dist	RWR	RWR	NetSimileSVM
NetSimile-Dist	NetSimile	RWR	RWR	Eigenvalues	Eigenvalues	BFS-Dist
RWR	RWR	IMIn&Known	Degree	BFS	BFS	RWR
BFS	RW	RWR-Dist	NetSimileSVM	AB	RW	BFS
Trans.	Degree	IMKnown	BFS	NetSimile	NetSimileSVM	RW

Table 4: Five methods that produced the closest rankings to the consensus ranking. NetSimile (or a variation) and RWR occur in every list.

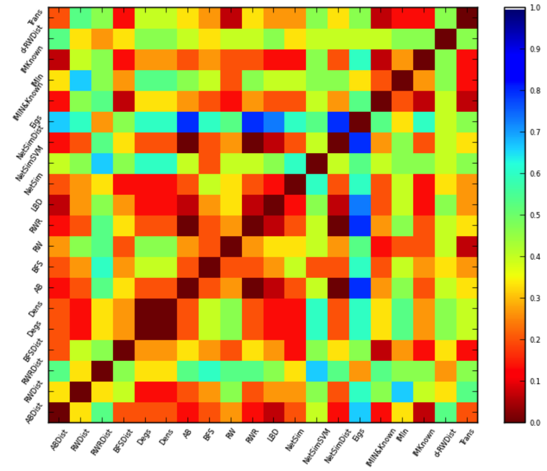

Figure 4: Kendall-Tau distances between rankings on network Twitter Retweets. Distances are typically below 0.5, indicating positive correlations. However, correlations are lower than seen on cross-sectional data.

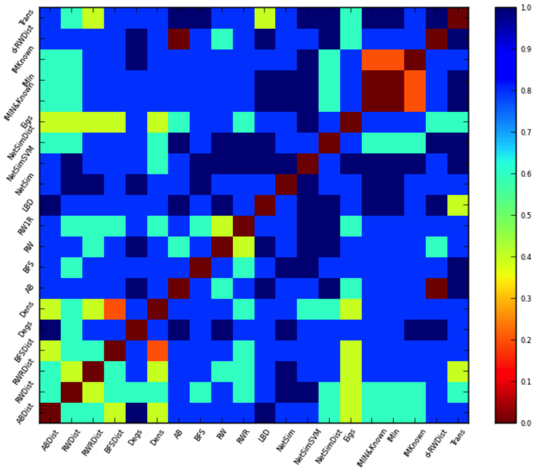

Figure 5: Overlap between top five ranked networks in each ranking on network Twitter Replies. Values indicate the fraction (out of 5) of elements that are shared between two rankings. Observe that overlaps are generally very small.

the longitudinal datasets are lower than on the cross-sectional data, we formulate two hypotheses. First, we consider the possibility that the ranking methods agree on the top ranked items (which are arguably the most important items), but disagree on the other items, leading to a low overall correlation. To analyze this, we calculate the overlap between the top-5 elements of each ranking. Figure 5 contains these results. Even when we only consider the top elements from each ranking, there is significant disagreement between the methods. Second, we consider the possibility that the time-step used to generate each of the networks (particularly on the daily datasets) may have been too small, resulting in networks that are unstructured sets of edges. Such lack of structure might make it impossible for a similarity method to produce a reasonable ranking. We consider this hypothesis in details next.

Dataset Aggregation. To explore our second hypothesis (i.e. the networks are just a set of dyads), we aggregate the three datasets on a larger time scale. For Yahoo! IM and Twitter Replies we aggregate the networks on a weekly basis and a three-day (for Twitter Replies, which had 30 days worth of data) or four-day

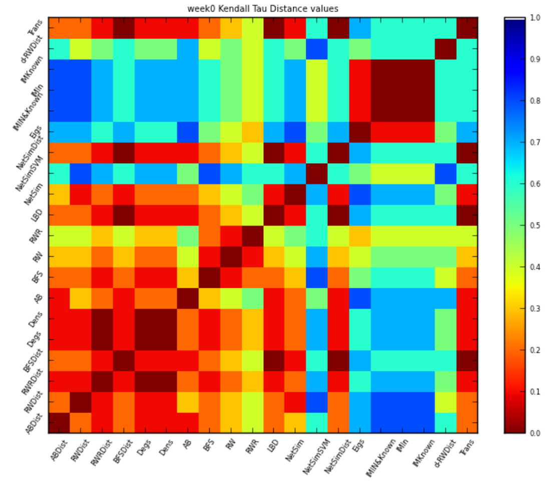

Figure 6: Kendall-Tau distances for network Twitter Replies aggregated on a weekly basis. Distances are much lower (and correlations much higher) than when the networks were aggregated on a daily basis.

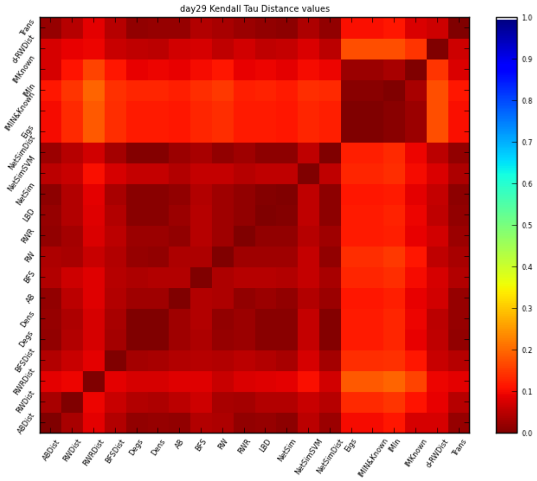

Figure 7: Kendall-Tau distances for network Twitter Replies aggregated on a vertical basis. Most distances are vey low, indicating almost perfect correlations.

(for Yahoo! IM, which had 28 days of data) basis. For these newly-aggregated datasets, we choose the reference graph by using the network that contains the reference network from the original dataset.

We also aggregate the networks vertically. For instance, the vertical datasets for networks Yahoo! IM and Twitter Replies contain 28 and 30 networks, respectively, where the first network contains the data from the first day, the second network contains the data from the first two days, the third network contains the network from the first three days, and so on. Twitter Retweets contains 5 networks, where the first contains the data from the first month, the second from the first two months, and so on. For these three vertical datasets, we select the reference graph to be the final network.

Figure 6 contains the Kendall-Tau distances between rankings obtained by aggregating the Twitter Replies data on a weekly basis. Results for Yahoo! IM, and those obtained by aggregating networks on a three- or four-day basis are similar. The correlations here are much higher than in the daily version of these datasets, suggesting that once a network has sufficient structure, the ranking methods will agree.

Figure 7 contains the Kendall-Tau distances over the vertically aggregated datasets for Twitter Replies. The correlations here are astoundingly high, indicating that the different methods are producing almost identical rankings. We see similar behavior on the Yahoo! IM vertical dataset, with the exception of d-RW-Dist, which behaves differently from the others (but is still positively correlated). On the Twitter Retweets vertically aggregated dataset, we see that 16 of the 20 methods are very highly correlated, while methods Eigs, IMIn, IMKnown, and IMIn&Known are different from the others, but very well-correlated with one another. We see hints of this behavior on the Twitter Replies vertically aggregated dataset as well.

Next, we again perform complete-linkage hierarchical clustering on these seven aggregated datasets. We see only two clusters that appear in more than half of the aggregated datasets: the cluster containing the three InfoMap methods, and the cluster containing the three InfoMap methods and Eigenvalues. Recall that these four methods are the only network-level methods.

Finally, we compute the Kemeny-Young consensus rankings. Figure 8 contains a heatmap depicting the Kendall-Tau distance of each method from the consensus, for each aggregated dataset. The NetSimileDist method, a variation of NetSimile, is consistently close to the consensus across these seven datasets. Interestingly, we saw that on our original cross-sectional experiments, some variation of NetSimile was also consistently close to the consensus across the different networks.

6 Discussion

Our results demonstrate that various network-similarity methods behave very similarly, even though they often have very different objective functions. On cross-sectional data, where differences between networks are clear, the different network similarity methods produce highly correlated rankings. On longitudinal datasets (such as Twitter Replies and Yahoo! IM), the methods were less correlated. When aggregating networks on a three- or four-day basis, or a weekly-basis, we once again observe higher correlations between rank-

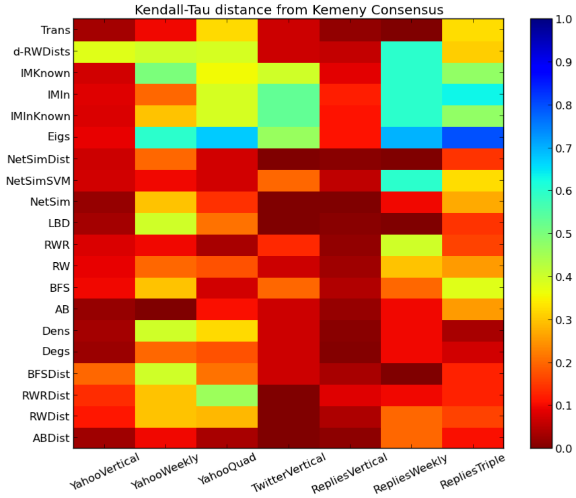

Figure 8: Kendall-Tau distances between methods and the Kemeny-Young consensus for aggregated datasets NetSimileDist is often close to the consensus.

ings. We also saw that correlations on the Twitter Retweets dataset, which was aggregated on a monthly-basis, were fairly high. We draw several conclusions from these results. First, use of a complex network similarity method is often unnecessary. We saw on the cross-sectional data that many complex methods, such as BFS-Dist, RW-Dist, and RWR-Dist, were highly correlated with a much simpler method, such as Density. In such cases, one can use the simpler, computationally efficient method as a substitute for the more costly methods. Second, it is critical to identify networks using data collected over an appropriate time interval. When examining networks aggregated daily, we saw that the network similarity methods produced very different rankings. We hypothesized that this was due partly to the lack of structure in networks that were too 'young,' and when we aggregated the same data over larger time-steps, saw a large increase in correlations. We are currently studying the problem of determining how to calculate sufficiently long time intervals, and are considering methods such as inspecting the degree distribution, triangle count, or diameter. Third, when networks are very similar, the biases of different network similarity methods emerge, resulting in lower correlations. When comparing networks that are different, one can use a simple method. When comparing networks that are very similar, such as different snapshots of the same network, selection of a single network similarity method becomes more challenging. In such a case, one can use the Kemeny-Young consensus as a summary of a variety of different rankings.

7 Conclusions

We introduced approaches for comparing and contrasting network-similarity methods. We conducted the first large-scale empirical study of network-similarity methods on both cross-sectional and longitudinal graphs. Our experiments demonstrated the following: (1) Different methods are surprisingly well correlated. (2) Some complex methods are closely approximated by much simpler methods. (3) A few methods produce similarity rankings that are very close to the consensus ranking. Moreover, we provided practical guidance in the selection of an appropriate network similarity method.

References

[1] Dblp. `http://dblp.uni-trier.de/`.

[2] Twitter. `http://www.twitter.com`.

[3] Yahoo. `http://webscope.sandbox.yahoo.com`.

[4] B. D. Abrahao, S. Soundarajan, J. E. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *KDD*, pages 624–632, 2012.

[5] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, pages 44–54, 2006.

[6] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Network similarity via multiple social theories. In *ASONAM*, 2013.

[7] J. He, J. Hopcroft, H. Liang, S. Suwajanakorn, and L. Wang. Detecting the structure of social networks using (α, β)-communities. In *WAW*, pages 26–37, 2011.

[8] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: structural role extraction and mining in large graphs. In *KDD*, pages 1231–1239, 2012.

[9] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:577–591, 1959.

[10] B. Klimt and Y. Yang. Introducing the enron corpus. In *CEAS*, 2004.

[11] D. Koutra, J. Vogelstein, and C. Faloutsos. DeltaCon: A principled massive-graph similarity function. In *SDM*, pages 162–170, 2013.

[12] G. N. Lance and W. T. Williams. Mixed-data classificatory programs i - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.

[13] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), 2007.

[14] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *WSDM*, pages 251–300, 2010.

[15] W. Richards and O. Macindoe. Decomposing social networks. In *SocialCom*, pages 114–119, 2010.

[16] M. Rosvall and C. T. Bergstrom. Maps of information flow on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.