# 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; 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 might examine more complex (and computationally burdensome) patterns such as communities.

We consider 20 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.

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

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344; and funded in part by NSF CNS-1314603 and DTRA HDTRA1-10-1-0120.

are surprisingly well correlated. (2) Some complex methods are closely approximated by much simpler methods. (3) A few methods produce similarity rankings that are 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. Sections 3 and 4 present our categorization of network-similarity methods and our comparison approaches. 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. 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 (e.g., DeltaCon [12] and most edit-distance based methods). Others do not require known node-correspondence (e.g., NetSimile [6] and graphlet-based approaches [16]).

## 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 3 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 3 types: *vector-based*, *classifier-based*, and *matching-based*. We describe 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 $1 - Canberra(F_1, F_2)$ [13].[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 describe each of these 20 methods 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. More details about these methods is available at `http://eliassi.org/graphcompareTR.pdf`.

Vector-based **NetSimile** [6] first calculates 7 local structural features for each node (backed by various social theories). It then calculates the median and the first four moments of distribution for each feature. These 5 statistics over the 7 features produce a length-35 "signa-

[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 $V$. [13]

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

Table 1: The twenty network-similarity methods considered in this paper categorized by (a) the level of network at which the method operates and (b) the type of comparison used.

ture" vector for the network. Classification-based **NetSimileSVM** samples 300 nodes from each network and calculates the 7 NetSimile local structural features for them. Matching-based **NetSimile-Match** uses the feature vectors obtained by NetSimileSVM. Vector-based **Random Walk Distances** (d-RW-Dist for short) performs 100 random walks of length $d$ (for $d = 10$, 20, 50, and 100) on each network. Each of these walks begins on a randomly selected node $u$ and ends on some other node $v$. It then calculates the shortest-path distance between $u$ and $v$ in the network. For each value of $d$, it aggregates these distances over all 100 random walks by calculating the median and first four moments of distribution for this set of values. Over all four values of $d$, it produces a length-20 feature vector. Vector-based **InfoMap-In (IMIn)**, **InfoMap-Known (IMKnown)**, **InfoMap-In&Known (IMIn&Known)** apply the Infomap community detection algorithm to the network [18]. For each node $n$, they identify which community $C$ that node $u$ 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 $u$, averaged over all nodes. IMIn&Known creates a length-2 feature vector containing both of these values. Classification-based **AB**, **BFS**, **RW**, **RWR** identify 300 communities on each network via the $\alpha$-$\beta$ swap algorithm [7], breadth-first-search, random walk without restart, and random walk with 15% chance of restart, respectively. For each of these communities, they calculate a length-36 feature vector including statistics such as conductance, diameter, density, etc. The full feature vector is described in [4]. Matching-based **AB-Match**, **BFS-Match**, **RW-Match**, **RWR-Match** use the same communities and feature vectors as identified by methods AB, BFS, RW, and RWR. Vector-based **Density**, **Degree**, **Transitivity** create a length-1 feature vector containing the density, average degree, or transitivity of each network. Vector-based **Eigenvalues (Eigs)** calculates the $k$ largest eigenvalues for each network. As in [6], we used $k = 10$. This defines a length-$k$ feature vector. Vector-based **LBD** computes 3 features from each network [17]. *Leadership* measures how much the connectivity of the network is dominated by one vertex. *Bonding* is simply the transitivity of the network. *Diversity* calculates the number of disjoint dipoles. Each network is represented by its length-3 vector.

## 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 confirm the results obtained by Kendall-Tau, we also calculate correlations using nDCG [9], which gives greater weight to elements

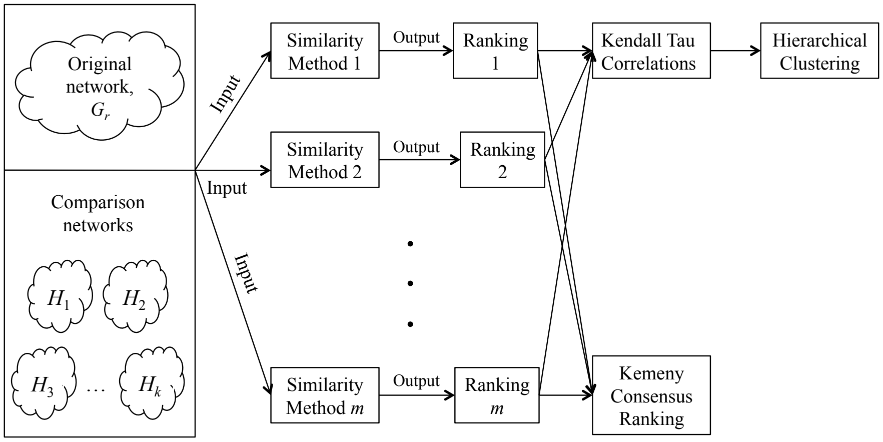

Figure 1: Flowchart of our two approaches. Both approaches use rankings generated by different similarity methods. In one (top) approach, we use the rankings to correlate and then cluster the network similarity methods. In the second (bottom) approach, we use the rankings to identify a single consensus ranking.

| **Network Name** | **# of Nodes** | **# of Edges** | **Avg. Degree** | **Max. Degree** | **Edge Density** | **Network Transitivity** | **# of CC** | **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 |

Table 2: Statistics for Our Cross-Sectional Networks. Observe the large variations in statistics across the different datasets. CC stands for connected components. LCC stands for largest connected component.

appearing near the beginning of the list. For each pair of methods, we calculate nDCG twice, using each of the rankings alternately as the 'true' ranking, and then average the results.

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. In our experiments, we used $l = 1000$. 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 [10]. 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

This section describes our datasets, methodology, and experiments on cross-sectional and longitudinal data.

| Network Name | # of Nodes | # of Edges | Avg. Degree | Max. Degree | Edge Density | Network Transitivity | # of CC | Frac. Nodes in LCC |
|---|---|---|---|---|---|---|---|---|
| Twitter Replies | 10K–27K | 7K–21K | 1.3–1.4 | 26–147 | $4.7 \times 10^{-5}$ – $13 \times 10^{-5}$ | 0.0–0.001 | 4K–9K | 0.007–0.05 |
| Twitter Retweets | 25K–120K | 28K–165K | 2.1–2.8 | 300–1300 | $2.3 \times 10^{-5}$ – $8.5 \times 10^{-5}$ | 0.02–0.03 | 3K–6K | 0.63–0.84 |
| Yahoo! IM | 28K–100K | 35K–180K | 2.5–3.6 | 66–123 | $3.6 \times 10^{-5}$ – $8.5 \times 10^{-5}$ | 0.08–0.20 | 600–3K | 0.48–0.85 |

Table 3: Statistics for Our Longitudinal Networks. Each dataset contains multiple networks, and so a range of values are presented for each statistic. Observe the large variations in statistics across different datasets.

**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. Tables 2 and 3 present 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 [15]. **DBLP**: a computer science coauthorship network [1]. **LJ1** and **LJ2**: portions of the LiveJournal blogging network [5]. **Enron**: the Enron e-mail network [11]. **Amazon**: a portion of the product co-purchasing network from Amazon.com [14].

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]. These three longitudinal datasets each exhibit very different structural characteristics. The Twitter Replies networks are typically very sparse and unstructured: on average, each connected component in these networks contains only 3 elements, and only 2% of the nodes appear in the largest connected component. In contrast, the Twitter Retweets and Yahoo! IM networks have more structure, with, respectively, an average connected component size of 12 and 22, and a largest connected component containing an average of 73% and 59% of the nodes.

**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 numerical similarity score.

**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. The average nDCG correlation (with 1 indicating the highest possible correlation) is 0.93, with a standard deviation of 0.06. Figure 2 contains the Kendall-Tau distances between the different methods for the case when DBLP was used as a reference graph. For brevity, we have omitted the heatmap depicting the nDCG correlations. Surprisingly, the different methods are usually correlated with one another even though they have different objective functions. Methods RW and RWR have an average Kendall-Tau distance across all networks of 0.09, and an average nDCG correlation of 0.99. 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 and an average nDCG of 0.99, despite having very different objective functions.

For each Kendall-Tau correlation calculation, we also calculate the accompanying p-value. Across all experiments, at p = 0.1, 50% of the methods have a significant correlation. Heatmaps showing the significant correlations are available at `http://eliassi.org/graphcompareTR.pdf`.

Next, we cluster the methods using complete-linkage hierarchical clustering on the pairwise Kendall-Tau distances. Here, we are interested in learning

| Cluster | Networks |
|---|---|
| IMIn&Known, IM Known | All networks: Grad, Undergrad, Amazon, DBLP, LJ1, LJ2, Email |
| RW-Match, RWR-Match | 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-Match, IMIn | 4 networks: Amazon, LJ1, LJ2, Email |
| RW-Match, RWR-Match, BFS-Match, Density | 4 networks: Amazon, LJ1, LJ2, Email |

Table 4: 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-Match | NetSimileSVM | AB-Match | RWR | RWR | NetSimileSVM |
| NetSimile-Match | NetSimile | RWR | RWR | Eigenvalues | Eigenvalues | BFS-Match |
| RWR | RWR | IMIn&Known | Degree | BFS | BFS | RWR |
| BFS | RW | RWR-Match | NetSimileSVM | AB | RW | BFS |
| Trans. | Degree | IMKnown | BFS | NetSimile | NetSimileSVM | RW |

Table 5: Five methods that produced the closest rankings to the consensus ranking. NetSimile (or a variation) and RWR occur in every list.

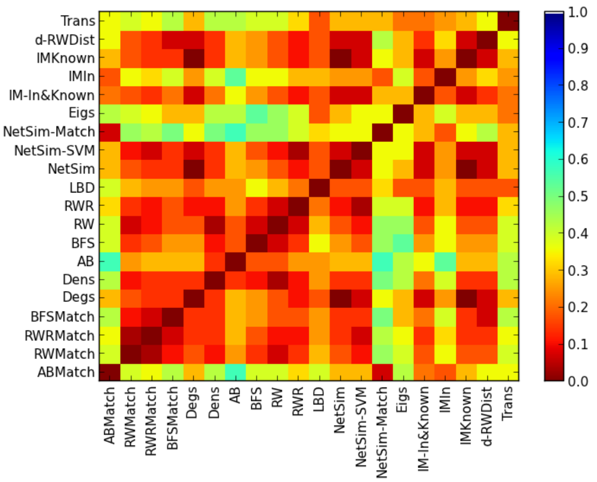

Figure 2: Heatmap showing Kendall-Tau distances between network similarity rankings when network DBLP was used as the reference network. Distances are generally low, indicating high correlations between rankings.

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 4 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-Match, RWR-Match, and BFS-Match 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 5 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 exhibited online 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

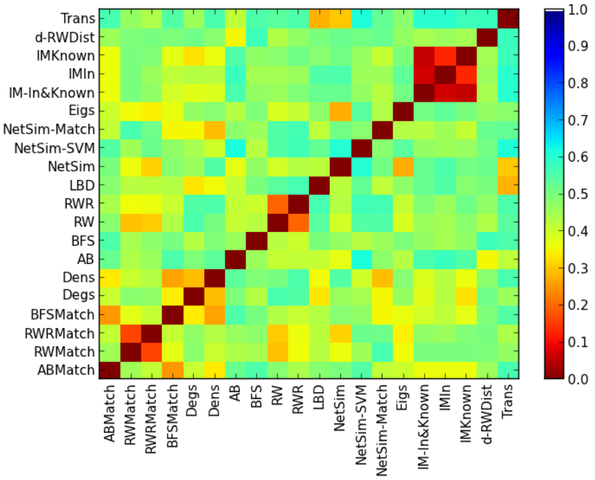

Figure 3: Kendall-Tau distances between rankings on network Twitter Replies. Observe that distances are very close to 0.5, indicating a lack of correlation.

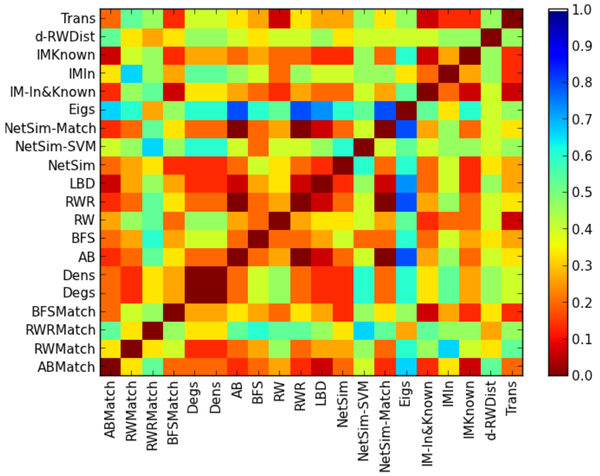

Figure 4: Kendall-Tau distances between rankings on network Twitter Retweets. Distances are typically below 0.5, indicating positive correlations. However, correlations are lower than seen on cross-sectional data.

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. When calculating correlations with nDCG, we again observe that Twitter Replies has the lowest correlations of all three datasets. On Twitter-Retweet, we observe an average nDCG correlation of 0.91 with a standard deviation of 0.06; on Yahoo! IM, we observe an average nDCG correlation of 0.87 with a standard deviation of 0.07; and on Twitter Replies, we observe an average nDCG correlation of 0.83 and a standard deviation of 0.06.

We calculate p-values for the Kendall-Tau correlations at the p = 0.1 level. On Twitter Retweets, 36% of the correlations are significant, on Twitter Replies, 23% are significant, and on Yahoo IM, 74% are significant. As with the cross-sectional data, plots showing these significance values are available at `http://eliassi.org/graphcompareTR.pdf`.

To answer the question of why correlations on 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 calculated the overlap between the top-5 elements of each ranking. Even when we only consider the top elements from each ranking, we observed significant disagreements 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 next.

**Dataset Aggregation.** We explore our second hypothesis (i.e., the longitudinal networks are just a set of dyads) by aggregating our three datasets on a larger time scale. For both Yahoo! IM, we aggregate its 28 networks on a weekly basis and on a four-day basis. For Twitter Replies, we aggregate its 30 networks on a weekly basis and on a three-day 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 cumulatively. For instance, the cumulative 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

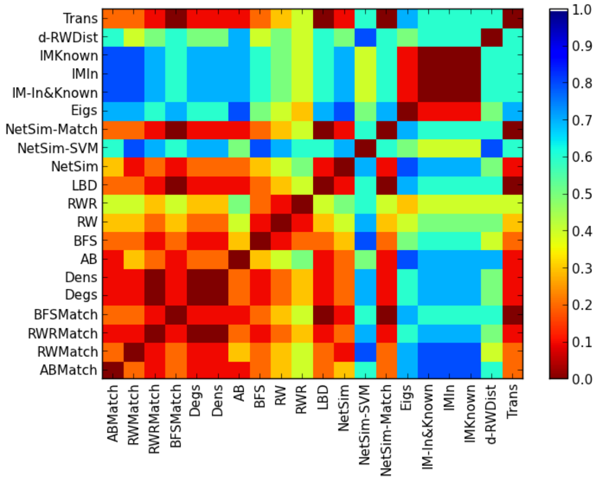

Figure 5: Kendall-Tau distances for network Twitter Replies aggregated on a weekly basis. Distances are much lower (and correlations much higher) than when the networks were aggregated on a daily basis.

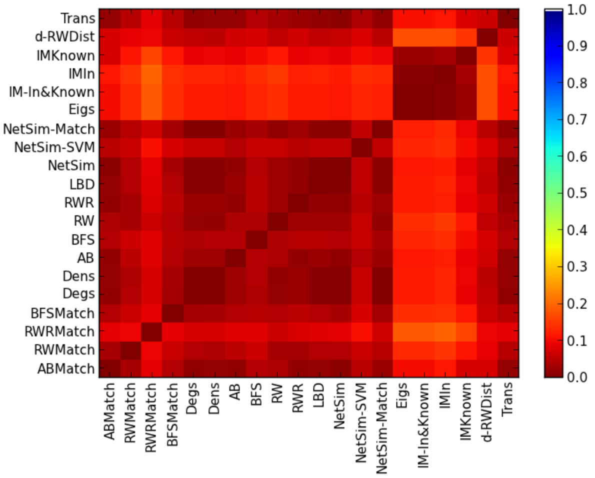

Figure 6: Kendall-Tau distances for network Twitter Replies aggregated on a cumulative basis. Most distances are vey low, indicating almost perfect correlations.

contains the data from the first month, the second from the first two months, and so on. For these three cumulative datasets, we select the reference graph to be the final network.

Figure 5 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 6 contains the Kendall-Tau distances over the cumulatively aggregated datasets for Twitter Replies. The correlations here are astoundingly high, and every correlation is significant at the $p = 0.1$ level, indicating that the different methods are producing almost identical rankings. We see similar behavior on the Yahoo! IM cumulative dataset, where 97% of the correlations are significant at $p = 0.1$. On the Twitter Retweets cumulatively aggregated dataset, we see that 63% are significant at $p = 0.1$, while methods Eigs, IMIn, IMKnown, and IMIn&Known are different from the others, but very well-correlated with one another.

Next, we again perform complete-linkage hierarchical clustering on these seven aggregated datasets. We observe 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 macro-level methods.

Finally, we compute the Kemeny-Young consensus rankings. Figure 7 contains a heatmap depicting the Kendall-Tau distance of each method from the consensus, for each aggregated dataset. The NetSimile-Match 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 rankings. 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, the use of a complex network similarity method is often unnecessary. We saw on the cross-sectional data that many complex methods, such as BFS-Match, RW-Match, and RWR-Match, 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.

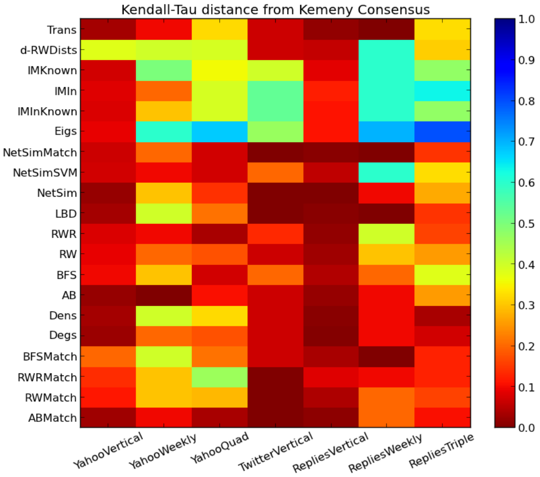

Figure 7: Kendall-Tau distances between methods and the Kemeny-Young consensus for aggregated datasets. NetSimile-Match is often close to the consensus.

Second, it is critical to identify networks using data collected over an appropriate time-interval. On 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' (i.e., were a set of dyads). When we aggregated the same data over larger time-steps, we observed 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.

Future directions for this work include extending the methods to other types of networks, including signed, directed, or weighted networks. The major challenge in this direction is generalizing existing network similarity methods for such networks.

## 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 $(\alpha, \beta)$-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. K. Kalervo Jarvelin. Cumulated gain-based evaluation of ir techniques. *ACM Trans. on Info. Sys.*, 20(4):422–446, 2002.

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

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

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

[13] G. N. Lance and W. T. Williams. Mixed-data classificatory programs I – Agglomerative systems. *Australian Computer J.*, 1(1):15–20, 1967.

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

[15] 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.

[16] N. Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23:177–183, 2007.

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

[18] M. Rosvall and C. T. Bergstrom. Maps of information flow on complex networks reveal community structure. *PNAS*, 105(4):1118–1123, 2008.