A Scalable Approach to Size-Independent Network Similarity

Michele Berlingerio[*] **Danai Koutra**[†] **Tina Eliassi-Rad**[‡] **Christos Faloutsos**[†]
[*]IBM Research Dublin [†]Carnegie Mellon University [‡]Rutgers University
[*]mberling@ie.ibm.com [†]{danai,christos}@cs.cmu.edu [‡]tina@eliassi.org

Abstract

Given a set of *k* networks, possibly with different sizes and no overlaps in nodes or edges, how can we quickly assess similarity between them, without solving the node-correspondence problem? Analogously, how can we extract a small number of descriptive, numerical features from each graph that effectively serve as the graph's "signature"? Having such features will enable a wealth of graph mining tasks, including clustering, outlier detection, visualization, etc.

We propose NETSIMILE – a novel, effective, and scalable method for solving the aforementioned problem. NETSIMILE has the following desirable properties: (a) It gives similarity scores that are size-invariant. (b) It is scalable, being linear on the number of edges for "signature" vector extraction. (c) It does *not* need to solve the node-correspondence problem. We present extensive experiments on numerous synthetic and real graphs from disparate domains, and show NETSIMILE's superiority over baseline competitors. We also show how NETSIMILE enables several mining tasks such as clustering, visualization, and discontinuity detection.

1 Introduction

We address the problem of *network similarity*. Specifically, given a set of networks of (possibly) different sizes, and without knowing the node-correspondences, how can we efficiently provide a meaningful measure of structural similarity (or distance)? For example, how structurally similar are the SDM and SIGKDD co-authorship graphs? How does their structural similarity compare with the similarity between the SDM and ICDM co-authorship graphs? Such measures are extremely useful for numerous graph-mining tasks. One such task is clustering: given a set of graphs, find groups of similar ones; conversely, find anomalies or discontinuities – i.e., graphs that stand out from the rest. Transfer learning is another application, if graphs G_1 and G_2 are similar, we can transfer conclusions from one to the other to perform across-network classification with better classification accuracy.

We define the **network similarity / distance** problem as follows. **Input:** A set of k anonymized networks of potentially different sizes, which may have no overlapping nodes or edges. **Output:** The structural similarity (or distance) scores of any pair of the given networks (or better yet, a feature vector for each network).[1]

The core of our approach, NETSIMILE, is a careful extraction and evaluation of structural features. For every graph G, we derive a small number of numerical features, which capture the topology of the graph as moments of distributions over its structural features. The similarity score between two graphs then is just the similarity of their "signature" feature vectors. Once we have the similarity function, we can do a wealth of data mining tasks, including clustering, visualization, and anomaly detection. Our empirical study includes experiments on more than 30 real-world networks and various synthetic networks generated by four different graph generators (namely, Erdös-Rényi,

[1]Throughout the paper, we assume similarity and distance are interchangeable.

Forest Fire, Watts-Strogatz, and Barabási Preferential Attachment). We compare NETSIMILE with two baselines. The first baseline extracts frequent subgraphs from the given graphs and performs pairwise comparison on the intersection of the two sets of frequent patterns. The second baseline computes the *k* largest eigenvalues of each network's adjacency matrix and measures the distance between them. Our experiments provide answers to the following questions: How do the various methods compare w.r.t. their similarity scores? Are their results intuitive (e.g., is a social network more similar to another social network than to a technological network)? How do they compare to null models? Are the methods just measuring the sizes of the networks in their comparisons? How scalable are the various methods? Can we build a useful taxonomy for networks based on their similarities?

Contributions

- *Novelty*: By using moments of distribution as aggregators, NETSIMILE generates a single "signature" vector for each graph based on the local and neighborhood features of its nodes.
- *Effectiveness*: NETSIMILE produces similarity / distance measures that are size-independent, intuitive, and interpretable.
- *Scalability*: The runtime complexity for generating NETSIMILE's "signature" vectors is linear on the number of edges.
- *Applicability:* NETSIMILE's "signature" vectors are useful in many graph mining tasks.

The rest of the paper is organized into the following sections: Proposed Method, Experiments, Related Work, and Conclusions.

2 Proposed Method

NETSIMILE has three steps. First, it *extracts* structural features from each given graph. This step produces a $node \times feature$ matrix per graph. Second, it *aggregates* each column of the $node \times feature$ matrix to produce a single "signature" feature vector per graph. Third, it *compares* the signature feature vectors of graphs. We discuss each of these steps below.

Feature extraction. NETSIMILE's feature extractor generates a set of structural features for each node based on its local and egonet-based features – a node's egonet is the induced subgraph of its neighboring nodes. Specifically, NETSIMILE computes the following features.

- $d_i = |N(i)|$: degree of node i; $N(i)$ denotes the neighbors of node i.
- c_i: clustering coefficient of node i, defined as the number of triangles connected to node i over the number of connected triples centered on node i.
- $\bar{d}_{N(i)}$: average number of node i's two-hop away neighbors, computed as $\frac{1}{d_i}\sum_{\forall j \in N(i)} d_j$.
- $\bar{c}_{N(i)}$: average clustering coefficient of N(i) , calculated as $\frac{1}{d_i}\sum_{\forall j \in N(i)} c_j$.
- $|E_{ego(i)}|$: number of edges in node i's egonet; $ego(i)$ returns node i's egonet.
- $|E^o_{ego(i)}|$: number of outgoing edges from $ego(i)$.
- $|N(ego(i))|$: number of neighbors of $ego(i)$.

Note that NETSIMILE is flexible enough to incorporate additional features. We choose these local and egonet-based features because they satisfy our constraints in terms of effectiveness (namely, size-independence, intuitiveness, and interpretability) and scalability (see Section 3).

Feature aggregation. After the feature extraction step, NETSIMILE has extracted a $node \times feature$ matrix, F_{G_j}, for each graph $G_j \in \{G_1, G_2, \cdots, G_k\}$. We can measure similarity between graphs by comparing their feature matrices (see discussion below). However, we discovered that generating a single "signature" vector for each graph produces more efficient and effective comparisons. To this end, NETSIMILE uses the following aggregators on each feature (i.e., on each column of F_{G_j}): *median*, *mean*, *standard deviation*, *skewness*, and *kurtosis*. Note that expect for median, the rest are moments of distribution of each feature. NETSIMILE is flexible enough to use other aggregators

as well, though we found these to be sufficient for the task of network comparison and satisfy our effectiveness and scalability constraints (see Section 3).

Comparison. After the feature aggregation step, NETSIMILE has produced a "signature" vector $\vec{s}_{G_j}$ for every graph $G_j \in \{G_1, G_2, \cdots, G_k\}$. NETSIMILE now has the whole arsenal of clustering techniques and pairwise similarity / distance functions at its disposal. Amongst the collection of pairwise similarity / distance functions, we found Canberra Distance ($d_{Can}(P,Q) = \sum_{i=1}^{d} \frac{|P_i - Q_i|}{P_i + Q_i}$) to be very discriminative (a good property for a distance measure). This is because Canberra Distance is sensitive to small changes near zero; and it normalizes the absolute difference of the individual comparisons.

Computational complexity. Let k = number of graphs given to NETSIMILE (i.e., $k = |\{G_1, \cdots, G_k\}|$), n_j = the number of nodes in G_j, m_j = the number of edges in G_j, f = number of structural features extracted, and r = number of aggregators used.

Lemma 1 *The runtime complexity for generating* NETSIMILE*'s "signature" vectors is linear on the number of edges in $\{G_1, \cdots, G_k\}$, and specifically*

$$O(\sum_{j=1}^{k}(fn_j + fn_j log(n_j))) \tag{1}$$

where $f \ll n_j \ll m_j$ and $n_j log(n_j) \approx m_j$ in real-world graphs.

Proof To generate NETSIMILE's "signature" vectors, structural features need to be extracted and then aggregated. The feature extraction part involves computing local and neighborhood-based structural features. As proved in [1], computation of neighborhood-based features is expected to take $O(n_j)$ for real-world graphs. Therefore to compute f neighborhood-based features on a graph G_j, it takes $O(fn_j)$. Feature aggregation takes $O(fn_j log(n_j))$ for each graph G_j. Recall that NETSIMILE's aggregators are median, mean, standard deviation, skewness, and kurtosis. The latter four can be computed in one-pass through the f feature values. The most expensive computation is the median which cannot be done in one-pass. However, it can be computed in $O(nlog(n) + n)$ for n numbers. Basically, one needs $O(nlog(n))$ to sort the n numbers. Then, a selection algorithm can be used to get the median with only O(n) operations. □

Remark: Network comparison through statistical hypothesis testing. Given the $node \times feature$ matrices of two graphs, F_{G_1} and F_{G_2}, NETSIMILE can use statistical hypothesis testing to see if the two graphs are samples from the same underlying distribution. Specifically, NETSIMILE normalizes each column (i.e. feature) in F_{G_1} and F_{G_2} by its L_2 norm. Then, NETSIMILE does pairwise hypothesis testing across the features of the graphs. For example, it does hypothesis testing between the degree columns in G_1 and G_2; between the clustering coefficient columns in G_1 and G_2; and so on. This process produces seven $p\text{-}values$ (corresponding to the seven features extracted by NETSIMILE). To decide whether the two graphs are from the same underlying distribution, NETSIMILE uses the maximum $p\text{-}value$. We also tried the average of the $p\text{-}values$, though that analysis did not produce as discriminative results as the maximum $p\text{-}value$.

For the statistical hypothesis tests, NETSIMILE can use any test available. We tried the Mann-Whitney Test [2] and the Kolmogorov-Smirnov Test [3]. The Mann-Whitney Test is nonparametric. It assumes two samples are independent and measures whether the two samples of observations have equally large values. The Kolmogorov-Smirnov Test is also nonparametric. We used the two-sample Kolmogorov-Smirnov Test which compares two samples w.r.t. the location and shape of the empirical cumulative distribution functions of the two samples. We found that neither test generated enough discriminative power to effectively capture differences between graphs (though the Mann-Whitney Test was more discriminative).

Remark: Network comparison at the local- vs. global-level. Whether one prefers local-level network similarity to global-level network similarity depends on the application for which the similarity is being used. NETSIMILE is designed such that it can take either local-level or global-level features. Here, we emphasis NETSIMILE's local-level network similarity. The advantages of local-level comparison is that node-level and egonet-level features are often more interpretable than global features – e.g., consider average degree of a node vs. the number of distinct eigenvalues of the adjacency matrix. Also, local-level features are computationally less expensive than global-level features –

e.g., consider clustering coefficient of a node vs. diameter of the graph. Moreover, looking at local-level features answers the question: "are the given two networks from similar linking models?" For example, consider the Facebook and Google+ social networks. Even though Google+ is a smaller network than Facebook, are its users linking in a similar way to the users of the Facebook network? In other words, is the smaller Google+ network following a similar underlying model as the lager Facebook network? Local-level features can capture any similarity present in the linking models of the two networks, but global-level features cannot.

3 Experiments

We ran experiments on over 30 real-world graphs including **arXiv:** 5 co-authorship networks from arXiv corresponding to different fields, **DBLP-C:** 6 co-authorship networks from DBLP corresponding to different conferences, **DBLP-Y:** 5 DBLP co-authorship networks spanning 2005 to 2009, **IMDB:** 5 collaboration networks from IMDB for movies issued from 2005 to 2009, **Query Log:** 5 word co-occurrence networks built from a query log of approximately 20M web-search queries submitted by 650K users over 3 months, **Oregon AS:** 5 autonomous systems routing graphs collected between March 31st and May 26th 2001. For experiments on synthetic graphs, we generated Barabási-Albert, Forest Fire, Erdös-Rényi, and Watts-Strogatz graphs based on different settings. Details are available in [4]. For each generator and for each node-set size, we built five networks. Our results (on synthetic graphs) report the average values obtained across the five networks per generator and node-set size. We implemented our approach in C++ and Matlab, making use of the GNU Statistic Libraries and igraph. The code was run on a server equipped with 8 Intel Xeon processors at 3.0GHz, with 16GB of RAM, and running CentOS 5.2 Linux.

The rest of this section is organized as follows. We describe our baseline methods next. Then, we present results that answer the following questions: How do the different approaches compare? Is there a particular method which clearly outperforms the others? If yes, to which extent? How can we interpret the results? Can we build a taxonomy over the networks based on our results? Is NETSIMILE affected by the sizes of the networks? How well does NETSIMILE perform in various graph mining applications?

3.1 Baseline Methods

We compare NETSIMILE with (a) Frequent Subgraph Mining and (b) Eigenvalues Extraction. We chose these two methods because they are intuitive and widely applicable. Many methods discussed in Section 4 are application-dependent.

FSM (Frequent Subgraph Mining): Given two graphs, we take the intersection of their frequent pattern-sets and build two vectors (one per graph) of relative supports of their patterns [5]. We compare these FSM vectors with NETSIMILE's "signature" vectors using Cosine Similarity and Canberra Distance. A clear drawback of FSM is its lack of scalability (since it relates to subgraph isomorphism).

EIG (Eigenvalues Extraction): This is an intuitive measure of network similarity that is based on *global* feature extraction (as opposed to the *local* feature extraction of NETSIMILE). For each graph, we compute the k largest eigenvalues[2] of its adjacency matrix, and thus we obtain a vector of size k per graph. Then, we use the Canberra Distance in order to compare these vectors and find the pairwise similarities between the graphs. A disadvantage of EIG is that it is size dependent: larger networks - or ones with larger LCC (Largest Connected Component) - have higher eigenvalues. Thus, EIG will lead to higher similarity between networks with comparable sizes. Moreover, there is no global upper-bound for eigenvalues, making distance values hard to compare.

3.2 Entropy of Graph Feature Vectors

We measure the entropy in feature vectors generated by NETSIMILE, FSM, and EIG on the DBLP-C co-authorship networks. As Figure 1 shows, NETSIMILE's feature vectors have higher entropy than FSM's or EIG's. Higher entropy means more uncertainty (i.e., we need more bits to store the

[2]We tried a few values for k and saw no significant changes around 10; so we selected $k = 10$.

desired information). So, NETSIMILE's feature vectors capture the nuances (i.e. uncertainty) in the graphs bettern than FSM or EIG, which leads to more discriminative power w.r.t. graph comparison.

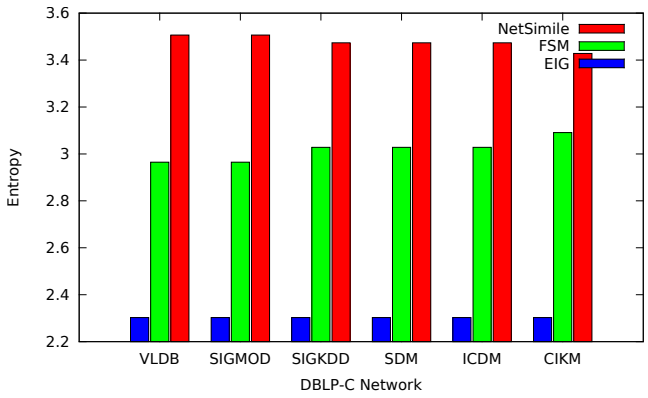


Figure 1: Entropy of feature vectors generated by NETSIMILE, FSM, and EIG on the DBLP-C co-authorship networks. NETSIMILE's feature vectors have higher entropy than FSM's or EIG's, which implies that they are capturing the nuances in the graphs better than FSM or EIG.

3.3 Interpretability of Results

To make sense of our results, we exploit the background knowledge about the networks used in our experiments. Amid the real networks, we have three sets of collaboration networks (DBLP-C, DBLP-Y and IMDb), one technological network (Oregon AS), and a word co-occurrence network (Query Log). In addition, we have different synthetic networks generated by various commonly used models. One would expect these networks to be "clustered" by their types. This idea was inspired by the considerations found in [6], where a large set of networks of different types are analyzed, together with their typical global and local features. For these experiments, we use agglomerative clustering [7] with Canberra Distance and unweighted average linking since it allows for easy interpretation of results.

Figure 2(a) presents the dendrogram of all of our networks built by hierarchical agglomerative clustering with unweighted average linking and the Canberra Distance and using NETSIMILE's graph "signature" vectors. The network names are colored by data set. As evident in Figure 2(a), there is a clear distinction between the clusters. The collaboration networks appear all together, along with the forest fire synthetic networks. The Oregon AS forms a cluster that only at the height of 0.45 joins with the Query Log. The Erdös-Rényi and Watts-Strogatz form a separate cluster. This, in turns, reflects our aforementioned intuition about following our background knowledge of the data.

Figure 2(b) shows the dendrogram for the above experiment (hierarchical agglomerative clustering with unweighted average linking and the Canberra Distance) for graph vectors generated by EIG. This figure clearly shows a different picture, where the networks are grouped differently (see how the distribution of the colors is mixed). For example, in the leftmost cluster, two collaboration networks from arXiv are put together with four Query Log networks, while the missing Query Log network is placed together with the Oregon AS networks. The EIG results are not intuitive, thus making EIG not suitable for interpreting graph-similarity results.

3.4 Similarity of Networks with Different Sizes

One question that may arise regarding NETSIMILE is whether its results are affected by the differences in sizes or other basic statistics of the two networks being compared. We do not want the size to play an important role in our solutions given that our interpretation of the question "are two networks similar?" leads to the question "do the two networks follow the same (or similar) underlying linking model?".

To answer the aforementioned questions, we compared the relationships between the NETSIMILE with Canberra Distance and some basic statistics of our real and synthetic networks. Specifically, we

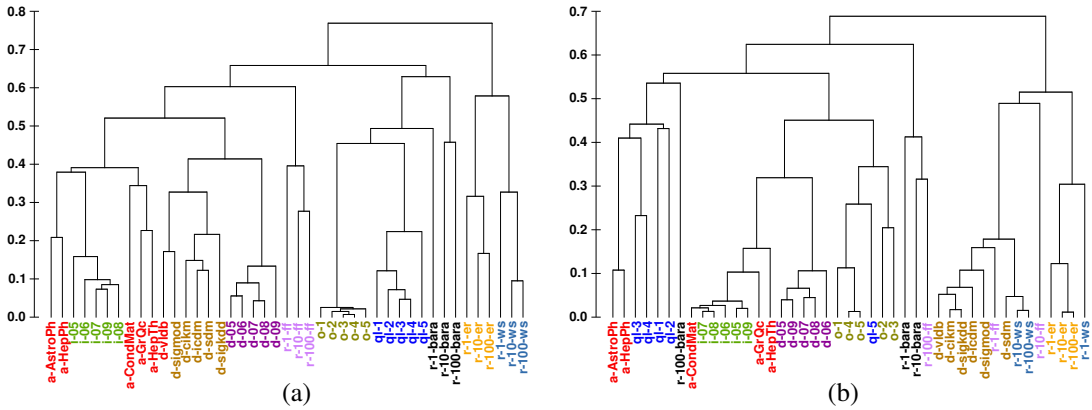
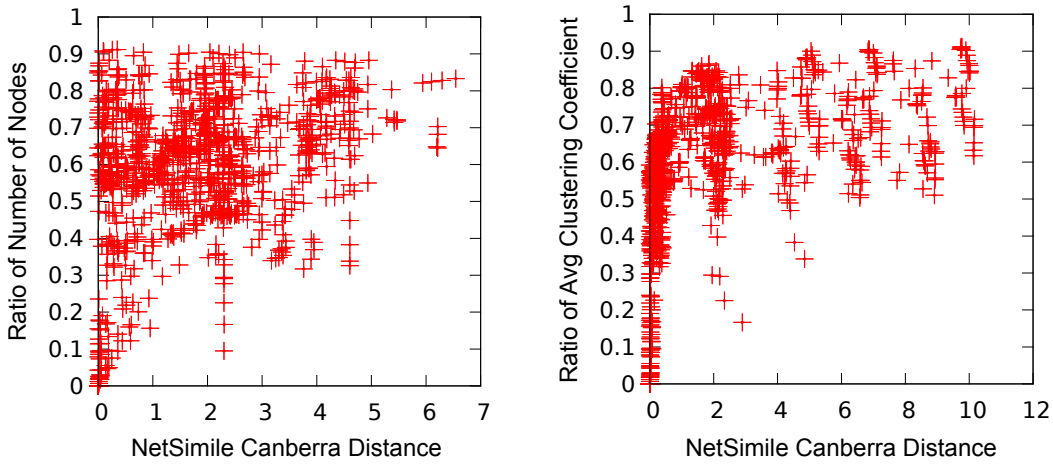


Figure 2: Hierarchical dendrograms of all network based on (a) NETSIMILE with Canberra Distance, and (b) EIG with Canberra Distance. Network names are colored by data set. Homogeneity in colors (NETSIMILE's dendrogram) indicates better and more intuitive groupings (than EIG's dendrogram).



(a) NETSIMILE vs. ratio of # nodes

(b) NETSIMILE vs. ratio of avg. clustering coefficient

Figure 3: NETSIMILE Canberra Distance is not measuring size, as there is no clear evidence of correlation between the two axes.

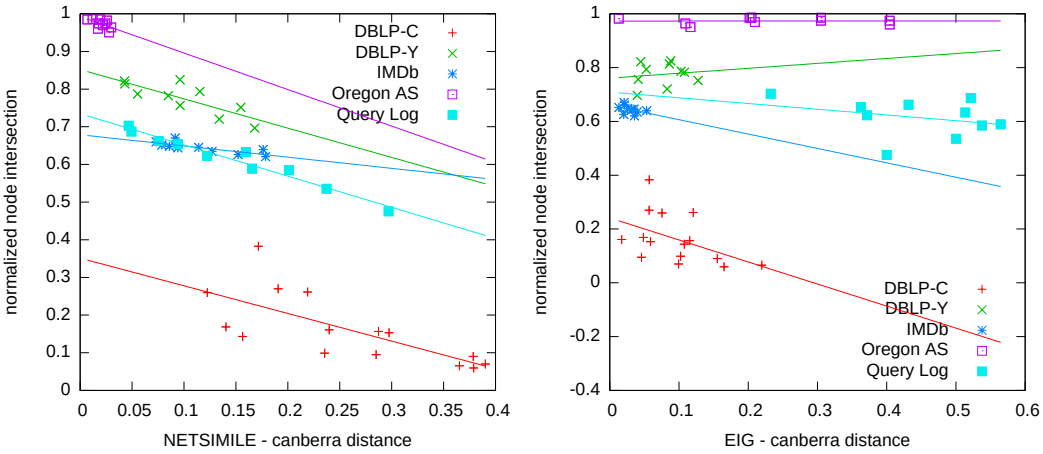
compared NETSIMILE values of two networks with the ratio between their (1) number of nodes, (2) number of edges, (3) average clustering coefficients of the nodes, (4) average degree, (5) maximum degree, and (6) network clustering coefficient. In all of them, we saw no correlation. For brevity, we only show the scatterplot for the NETSIMILE values and the ratio between the number of nodes of the two networks (see Figure 3(a)) and the scatterplot for the NETSIMILE values and the ratio between the average clustering coefficients of the nodes of the two networks (see Figure 3(b)). As evident in these scatterplots, NETSIMILE's results are not merely reflecting the difference in sizes of the networks. If they were, we would expect to observe correlations among the points in each scatterplot. This implies that we can generate two networks of the same kind, with different sizes (e.g., Forest-Fire networks [8] with 10K and 100K nodes) and NETSIMILE would find them similar.

3.5 Applications

NETSIMILE can be used in numerous graph mining applications. Here we discuss one of them. See [4] for more applications.

NETSIMILE as a Measure of Node-Overlap. Given three graphs G_A, G_B, and G_C of the same domain (e.g., co-authorship networks in *SIGMOD*, *VLDB* and *ICDE*), can we use *only* their NET-

SIMILE's "signature" vectors to gauge the amount of node-overlap between them? Our hypothesis is that if graph G_A is more similar to graph G_B than graph G_C, then G_A will have more overlap in terms of nodes with G_B than G_C. To test this hypothesis, we ran NETSIMILE with Canberra Distance on our real networks. Figure 4(a) depicts the scatterplot of NETSIMILE results on graphs within each comparable group (i.e., arXiv, DBLP-C, DBLP-Y, IMDb, Query Log, and Oregon AS graphs). The y-axis is the normalized node overlap and is equal to $\frac{|V_{G_A} \cap V_{G_B}|}{\sqrt{|V_{G_A}| \times |V_{G_B}|}}$. As the figure shows the lower the NETSIMILE Canberra Distance, the higher the normalized node intersection. This confirms our hypothesis that NETSIMILE can be used to gauge node-overlap between two graphs without node correspondence information. Figure 4(b) shows the same scatter plot, but computed using the EIG Canberra Distance approach. In this case, there is no correlation between node overlap and the distance. Due to its scalability issues, the FSM approach could not be computed on all the networks in Figure 4.



(a) (b)

Figure 4: (a) NETSIMILE Canberra Distance on DBLP, IMDb, Oregon and QueryLog. (b) EIG Canberra Distance on the same networks. NETSIMILE is an effective measure for node overlap without any node-correspondence information. The lower the NETSIMILE Canberra Distance, the higher the normalized node intersection. This correlation does not hold for EIG. The points in both plots are along the fitted lines. For NETSIMILE (a), the root mean square of residuals are $6.5E{-}2$ for DBLP-C, $2.6E{-}2$ for DBLP-Y, $9.0E{-}3$ for IMDb, $1.4E{-}2$ for Oregon AS, and $6.5E{-}2$ for Query Log. For EIG (b), the root mean square of residuals are $8.2E{-}2$ for DBLP-C, $4.2E{-}2$ for DBLP-Y, $1.3E{-}3$ for IMDb, $1.2E{-}2$ for Oregon AS, and $6.7E{-}2$ for Query Log.

NETSIMILE as a Network Labeler. Given a new (*never before seen*) graph, can we use the Canberra Distance between its NETSIMILE's "signature" vector to known graphs' NETSIMILE "signature" vectors to accurately predict its label? To answer this question, we setup and ran the following 4-step experiment. In step 1, we created a set of *test* graphs by generating 50 synthetic graphs of types Erdös-Rényi, Watts-Strogatz, Barabási, and Forest Fire. In step 2, for each test graph, we compared its NETSIMILE score using the normalized Canberra Distance with our real-world graphs. In step 3, we assigned to the test graph the label of its most similar graph. In step 4, we computed the accuracy of our predictions.

The predictive accuracy of NETSIMILE was 100% – i.e., NETSIMILE was able to label all 50 test graphs accurately. For each of the 50 test graphs, we inspected the NETSIMILE normalized Canberra Distance between the most similar graph (whose label we chose) and the second most similar graph (whose label we did not choose). Let's call the former $dist_1$ and the latter $dist_2$. The minimum difference between $dist_1$ and $dist_2$ across the 50 test graphs was 0.001. The maximum was 0.428. The mean difference was 0.143; and the standard deviation was 0.112. Thus, the answer to the aforementioned question of whether NETSIMILE can be used effectively as a network labeler is yes. We ran the same experiments using EIG with Canberra Distance on the same networks. The predictive accuracy of EIG was 72%, i.e., 14 graphs were uncorrectly labeled.

4 Related Work

Assessing the similarity between two "objects" comes up in numerous settings. Thus, the literature is rich in similarity measures for various domains: distributions or multi-dimensional points [9], datacubes [10], and graphs, such as social ([11, 12]), information [13], and biological networks [14]. Berlingerio et al. [4] provide a detailed account on graph similarity when the node correspondences are unknown. Our work is different from previous work because it uses moments of distributions of local structure; and in this way is able to measure network similarity in a scalable, size-independent, intuitive, and interpretable.

5 Conclusions

We introduced NETSIMILE, a novel, effective, size-independent, and scalable method for comparing large networks. NETSIMILE has three components: (1) feature extraction, (2) feature aggregation, and (3) comparison. The heart of our contribution is in components (1) and (2), where we discovered that moments of distributions of structural features computed on the nodes and their egonets provide an excellent "signature" vector for a graph. These "signature" vectors can be used to effectively and quickly assess the similarity of two or more graphs. Our broader contributions are as follows. *Novelty*: NETSIMILE avoids the (expensive) node correspondence problem, as well as adjusts for graph size. *Effectiveness*: NETSIMILE gives results that agree with intuition and the ground-truth. *Scalability*: NETSIMILE generates its "signature" vectors in time linear on the input size (i.e., number of edges of the input graphs). *Applicability*: NETSIMILE's "signature" vectors are useful in numerous graph mining tasks. In addition, NETSIMILE is easily extensible to include features and aggregators besides the ones presented.

References

[1] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It's who you know: Graph mining using recursive structural features. In *KDD*, pages 663–671, 2011.

[2] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables in stochastically larger than the other. *AMS*, 18(1):50–60, 1947.

[3] M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *JASA*, 69(347):730–737, 1974.

[4] Michele Berlingerio, Danai Koutra, Tina Eliassi-Rad, and Christos Faloutsos. NetSimile: A scalable approach to size-independent network similarity. *CoRR*, abs/1209.2684, 2012.

[5] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *ECML PKDD*, pages 115–130, 2009.

[6] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, pages 520–528. Springer, 5th edition, 2011.

[8] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187, 2005.

[9] Sung-Hyuk Cha. Comprehensive survey on distance / similarity measures between probability density functions. *Int'l J. of Mathematical Models & Methods in Applied Sciences*, 1(4):300–307, 2007.

[10] Eftychia Baikousi, Georgios Rogkakos, and Panos Vassiliadis. Similarity measures for multidimensional data. In *ICDE*, pages 171–182, 2011.

[11] Katherine Faust. Comparing social networks: Size, density and local structure. *Advances in Methodology and Statistics*, 3(2):185–216, 2006.

[12] Owen Macindoe and Whitman Richards. Graph comparison using fine structure analysis. In *IEEE Int'l Conf. on Privacy, Security, Risk and Trust*, pages 193–200, 2010.

[13] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. Web graph similarity for anomaly detection. In *WWW*, pages 1167–1168, 2008.

[14] Haiyan Hu, Xifeng Yan, Yu Huang, Jiawei Han, and Xianghong Jasmine Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21:213–221, 2005.