Predicting Graph Categories from Structural Properties

James P. Canning
SUNY Geneseo

Emma E. Ingram
University of Alabama

Sammantha Nowak-Wolff
Valparaiso University

Adriana M. Ortiz
University of Puerto Rico

Nesreen K. Ahmed
Intel Labs

Ryan A. Rossi
Adobe Research

Karl R. B. Schmitt
Valparaiso University

Sucheta Soundarajan
Syracuse University

ABSTRACT

Complex networks are often categorized according to the underlying phenomena that they represent such as molecular interactions, re-tweets, and brain activity. In this work, we investigate the problem of predicting the category (domain) of arbitrary networks. This includes complex networks from different domains as well as synthetically generated graphs from five different network models. A classification accuracy of 96.6% is achieved using a random forest classifier with both real and synthetic networks. This work makes two important findings. First, our results indicate that complex networks from various domains have distinct structural properties that allow us to predict with high accuracy the category of a new previously unseen network. Second, synthetic graphs are trivial to classify as the classification model can predict with near-certainty the network model used to generate it. Overall, the results demonstrate that networks drawn from different domains (and network models) are trivial to distinguish using only a handful of simple structural properties.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; *Combinatorics*; *Graph theory*; • **Information systems** → **Data mining**; • **Computing methodologies** → **Machine learning**; **Artificial intelligence**; **Logical and relational learning**; • **Networks** → Network types;

KEYWORDS

Network classification, network categorization, graph classification, graph features, structural properties, across-domain graph classification, network science, complex networks.

1 INTRODUCTION

Networks are often categorized according to the underlying phenomena that they represent, such as re-tweets, brain activity, or web page links. While there are inherent commonalities in network structure across different domains (*e.g.*, a power law degree distribution), it is also generally believed that networks from different categories have inherently unique network characteristics. In this work, we find strong evidence supporting this hypothesis by learning a multiclass classification model $f : \mathbf{x} \rightarrow y$ that is able to *accurately* predict (with 96.6% accuracy) the category of a new arbitrary network G' described only by a D-dimensional feature vector $\mathbf{x}'$ where $y \in \{1, 2, \ldots, K\}$ is the class label representing the category of a graph, *i.e.*, domain of a complex network or network model of a synthetically generated graph. The multiclass classification model f is learned using 785 networks from $K = 13$ categories (See Figure 1) which are characterized using only $D = 12$ simple structural features, such as average degree, total number of triangles, and edge density. We also investigate a classification model that uses only $D = 4$ features for predicting the category of unknown networks. The structural features were selected since they are computationally efficient to compute for large networks while also being the most basic fundamental properties of networks that allow us to accurately predict the category (domain) of a previously unseen network. Obviously, more complex structural features such as those based on graphlets (network motifs) [2, 36] are likely to further improve the accuracy. However, such complex structural features are computationally expensive to compute, but most importantly, the results in this work indicate that they are not needed to accurately predict the categories (domains) of networks. In other words, we observe that networks from different domains can be accurately distinguished using only the most basic and fundamental structural properties. The findings of this work were first published in September 2017 as a short paper, see [14].

Most previous research has focused on either (i) classification of synthetic graphs [11] or (ii) graphs within a particular category (domain) such as molecular graphs [30, 41, 54]. Other examples include distinguishing between brain or breast cancer cells [32] or distinguishing between different social structures [51]. One of the challenges of network classification is collecting a sufficient amount of data to classify. For this reason, most work on network similarity and graph classification have used synthetically generated graphs [10], as these can easily be created and customized. Alternatively, research using real-world networks has largely used graphs from the same domain such as chemical compounds or protein interactions [24, 32]. In those domains, generating a large number of graphs from the similar phenomenon is still relatively simple. Central to both resesarch themes is the investigation of new or different similarity measures. A review of many different similarity measures and their motivations can be found in [50].

In this work, we investigate the problem of predicting the domain (category) of arbitrary networks using a small set of graph features. This allows us to study questions such as whether network categories are distinguishable from one another (using both real complex networks from a variety of domains and synthetic graphs from network models), and which network properties are most

ACTUAL \ PREDICTED	Brain	Chem	Eco.	FB	RT	Road	Soc.	Web	BA	CL	ER	KPGM	SW	Recall
Brain	30	1	2	1	1	0	0	0	0	0	0	1	0	0.83
Chem	0	119	0	0	0	0	0	0	0	0	0	0	0	1
Ecology	0	0	5	0	0	0	1	0	0	0	0	0	0	0.83
Facebook	0	0	0	112	0	0	0	0	0	0	0	0	0	1
Retweet	0	0	0	0	58	0	3	0	0	0	0	0	0	0.95
Road	0	0	0	0	0	4	0	0	0	0	0	0	1	0.80
Social	0	0	0	1	1	0	39	1	0	3	0	0	1	0.85
Web	0	0	0	0	0	0	8	8	0	1	0	0	0	0.47
Barabasi	0	0	0	0	0	0	0	0	50	0	0	0	0	1
Chung-Lu	0	0	0	0	0	0	0	0	0	75	0	0	0	1
Erdős–Rényi	0	0	0	0	0	0	0	0	0	0	75	0	0	1
KPGM	0	0	0	0	0	0	0	0	0	0	0	75	0	1
Small-world	0	0	0	0	0	0	0	0	0	0	0	0	108	1
Precision	1	0.99	0.71	0.98	0.97	1	0.76	0.89	1	0.95	1	0.99	0.98	

Figure 1: Classification results. The classification model is able to accurately predict with 96.6% accuracy the category/domain of arbitrary unknown networks. These results are from a random forest classifier, however, similar results were obtained with other base classifiers.

useful for distinguishing the categories. To answer this question we learn a random forest classifier using real and synthetic networks and use it to predict the domain of new previously unseen networks. Using this model, we achieve a classification accuracy of 96.6% for predicting the domain (or network model) of both real complex networks and synthetically generated graphs. Overall, the results indicate that networks drawn from different domains and network models are trivial to distinguish using only a handful of simple structural properties. Additionally, the classification models also highlighted networks that are outliers within their own categories, suggesting new potential directions for understanding those networks.

This work makes two important findings:

(1) Real-world networks from various domains have distinct structural properties that allow us to predict with high accuracy the category of an arbitrary network.
(2) Synthetic graphs are trivial to classify as the classification model can predict with near-certainty the network model used to generate the synthetic graph.

2 MATERIALS AND METHODS

This section presents the approach and experimental setup, including a description of the large collection of graph datasets and their categories, the graph features derived from the networks, and the models used for prediction.

2.1 Network collections & data

Data was obtained from the Network Repository (NR) [43] for all non-synthetic graphs.[1] This included 1241 graphs. Of the network categories included on NR, three were from computational and algorithmic challenges (DIMACS, DIMACS10 and BHOSLIB) and two recorded graphs over time (temporal reachability, dynamic networks). As all five of these categories are fundamentally different from static networks from a discipline or field they were discarded as outside the problem scope. Finally, the cheminformatics category (containing graphs describing chemical bonds between atoms) had significantly more instances than all other categories and therefore was downsampled to 119 networks which is comparable to the 2nd largest category. In addition to this large collection of real-world networks, we also generated 383 synthetic graphs. These synthetic graphs were generated from 5 different graph models including:

- 75 using the Chung-Lu (CL) graph model [15]
- 75 using the Kronecker Product Graph Model (KPGM) [31]
- 108 using the Watts-Strogatz small-world (SW) graph model [55]
- 50 using the Barabási-Albert (BA) model [6]
- 75 using the Erdős-Rényi (ER) model [18]

These five different graph models and the specific parameters used to generate graphs from each of them are described below. The final classification data set has 785 graphs from 13 categories. A list of the network categories are provided in Figure 1.

Data Availability & Exploration: The network classification data used in this study is accessible online:

http://networkrepository.com/data/nc.csv

We also created an interactive graph visual analytics tool [4] to explore the network classification data in real-time over the web. This tool can be accessed at:

http://networkrepository.com/network-classification.

2.2 Synthetic graph models & settings

We describe the 5 different graph models used in this work below and discuss the graph model parameters used for each graph model.

Chung-Lu Graph Model: The Chung-Lu (CL) graph model [15] generates a synthetic graph with a given expected degree sequence.

[1] http://networkrepository.com/

Given a vector of expected degrees $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}$, an edge is created between node i and j with probability

$$p_{ij} = \frac{w_i w_j}{\sum_k w_k} \tag{1}$$

The expected degrees are based on the power law model with exponent θ. We generate CL graphs with the following parameters: $\theta \in \{1.7, 1.8, 1.9, 2.0, 2.1\}$ and

$$n \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$$

is the number of nodes.

Kronecker Product Graph Model: For the Kronecker product graph model (KPGM) [31], we follow the same methodology as described in [23]. In particular, the initiator matrix used is: $\left[\begin{smallmatrix} 0.57 & 0.19 \\ 0.19 & 0.05 \end{smallmatrix}\right]$. The number of nodes is $n = 2^k$ where $k \in \{8, 10, 12, 14, 16\}$ and the number of edges is $\alpha \cdot n$ where $\alpha \in \{8, 10, 12, 14, 16\}$. We repeat each combination of k and α three times to generate a total of 75 Kronecker graphs.

Watts-Strogatz small-world graphs: We also use synthetic graphs generated by the Watts-Strogatz small-world graph model [55]. This model creates a ring over n nodes then joins each node to its k nearest neighbors. Edges are randomly rewired with a constant probability p. For these graphs, we use $n \in \{100, 1,000, 10,000\}$, $k \in \{3, 4, 5, 6\}$, and randomly rewire the edges with $p \in \{0.1, 0.2, 0.3\}$. We repeat each combination of parameters (n, k, p) three times to generate a total of 108 small-world networks.

Barabasi-Albert preferential attachment graphs: The Barabasi-Albert (BA) preferential attachment model [6] matches expected scale-free degree distributions. The BA graph model starts with a connected network of one or more nodes and then adds nodes one at a time such that each new node is connected to σ existing nodes with a probability proportional to the number of links already existing in the graph. Thus, the new node has a preference to connect up to nodes that already have large degrees. More formally, the probability p_i of a new node forming an edge with node i is:

$$p_i = \frac{k_i}{\sum_j k_j} \tag{2}$$

where k_i denotes the degree of node i and $\sum_j k_j$ denotes the sum of degrees from all nodes that currently exist in the graph. We generated 25 BA graphs with 1,000 nodes and another 25 with 10,000 nodes for a total of 50 BA graphs. To select the number of edges σ to create between a new node and the existing nodes in the graph, we examined the the average degrees and number of edges in the real-world data and chose values that would make the synthetic BA networks "blend" in with the real-world networks. We generated 25 BA graphs with 1,000 nodes using $\sigma \in \{10, 40, 60\}$ and another 25 BA graphs with 10,000 nodes using $\sigma \in \{40, 60, 100\}$. Graphs with 100,000 nodes were excluded due to limits on computational resources available.

Erdős-Rényi graph model: Let $\textsc{er}(n, p)$ denote an Erdős-Rényi (ER) [18] graph that arises from fixing n nodes and generating edges independently with probability p. Thus, the expected degree for each node is simply $p(n-1)$. We generate three sets of 25 Erdős-Rényi graphs such that each set of 25 graph has a different number of nodes, that is, $n \in \{1000, 10000, 100000\}$. This gives a total of 75 ER graphs. To select the probability p that an edge exists between two nodes in the ER model, we looked at the densities of different sizes of graphs and chose p such that the resulting ER graph would have a similar density to the real-world networks used in this study. For graphs with 1,000 nodes we used $p \in \{0.05, 0.1, 0.2\}$; for graphs with 10,000 nodes we used $p \in \{0.0005, 0.005, 0.001\}$; and for graphs with 100,000 nodes we used $p \in \{0.0005, 0.00005, 0.000005\}$.

2.3 Graph features

In this work, we are interested in finding the simplest graph features that allow us to predict with high accuracy the domain (category) of each network data set. We represent each graph dataset using only $D = 12$ simple structural features. The features used for classification are defined below. Although one could use more complex features, such as graphlet-based features [2], we find that the simple properties that we consider are sufficient to achieve a high classification accuracy. Nevertheless, our results do not depend on the use of these more complex features. All graph features are normalized appropriately using min-max scaling before using them to train the classification models. In particular, each D-dimensional feature vector $\mathbf{x}$ is scaled as follows:

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \tag{3}$$

This ensures the feature values in $\hat{\mathbf{x}} \in \mathbb{R}^D$ are between zero and one. Similar results were observed using other norms. As an aside, we also tried normalizing each feature independently and did not observe any obvious benefit.

Graph feature definitions: The definitions of the graph features used in this work are provided below [37, 52]. Let $G = (V, E)$ be a graph with $|V|$ nodes and $|E|$ edges. Further, let $\Gamma_i = \{\, j \mid (i, j) \in E \,\}$ denote the set of nodes adjacent to node $i \in V$ and $d_i = |\Gamma_i|$ is the degree of node i.

- Density: The density of a graph G denoted as $\rho(G)$ is the ratio of edges in the graph to the amount of possible edges.
- Maximum degree: The maximum degree is the largest node degree in G defined as $\Delta(G) = \max\{d_1, d_2, \ldots, d_{|V|}\}$ where d_i is the degree of node $i \in V$, *i.e.*, the number of nodes adjacent to node i in the graph (neighbors of node i).
- Minimum degree: The minimum degree in G is defined as $\delta(G) = \min\{d_1, d_2, \ldots, d_N\}$. If there are nodes not connected to any other, the minimum degree is 0.
- Average degree: The average degree over all nodes in a graph G is defined as $d_{\text{avg}} = \frac{1}{|V|} \sum_i d_i$ where $d_i = |\Gamma_i|$ is the degree of node i.
- Assortativity coefficient: The assortativity coefficient captures the tendency of nodes to connect to other nodes with similar degree, or in contrast, the tendency of dissimilar nodes to connect [38]. More formally, the assortativity coefficient of a graph G is defined as

$$r(G) = \frac{|E|^{-1} \sum_{(i,j)\in E} d_i d_j - \left[|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}(d_i + d_j)\right]^2}{|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}(d_i^2 + d_j^2) - \left[|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}(d_i + d_j)\right]^2} \tag{4}$$

  where d_i and d_j are the degrees of the nodes at the ends of the edge $(i, j) \in E$. The summations in Eq. 4 are obviously over the set of edges E and thus is linear in the number of edges taking $O(|E|)$ time to compute.

- Total triangles: A triangle is a complete subgraph with exactly three vertices (3-clique). The total number of triangles in a graph G is the sum of all such triangles in G defined as $T(G) = \frac{1}{3}\sum_{e=(i,j)\in E} |\Gamma_i \cap \Gamma_j|$.
- Average triangles: Average number of triangles formed by the edges in G. More formally, let T_e denote the number of triangles containing edge $e = (i, j) \in E$, then $T_{\text{avg}} = \frac{1}{|E|}\sum_{e\in E} T_e$.
- Maximum triangles: The maximum number of triangles centered at any edge in the graph G defined as $T_{\max} = \max_{e\in E} T_e$ where $T_e = |\Gamma_i \cap \Gamma_j|$ is the number of triangles containing edge $e = (i, j) \in E$.
- Average clustering coefficient: The clustering coefficient of a graph quantifies how a node in a graph tends to cluster together [55]. More formally, the local clustering coefficient of a node $i \in V$ is $C_i = T_i/W_i$ where T_i is the number of triangles centered at node i and $W_i = d_i(d_i - 1)/2$ (paths of length two centered at i). Thus, the average local clustering coefficient of G is defined as $C(G) = \frac{1}{N}\sum_{i\in V} C_i$.
- Fraction of closed triangles (global clustering coefficient) [39]: Let $T(G)$ denote the number of triangles in G and let $W(G)$ denote the number of wedges (two-star paths), then the global clustering coefficient (density of triangles in G) is defined as $\kappa(G) = T(G)/W(G)$.
- Maximum k-core: A k-core of G is a maximal subgraph of G such that for all vertices in the subgraph, the degree is greater or equal to k. The maximum k-core of G is the largest k and denoted by $K(G)$.
- Maximum clique (lower-bound): The maximum clique size is defined as $\omega(G) = \max\{|P| : \text{P is a clique in } G\}$ where P is a clique in G such that every pair of vertices $(u, w) \in P$ are connected by an edge forming a complete subgraph. In this work, we use a heuristic clique finder that has been shown to often find the largest such clique in G [46].

2.4 Models

Given N training graphs $\{\mathbf{x}_i, y_i\}_{i=1}^N$ where each $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional feature vector for G_i and $y_i \in \{1, 2, \ldots, K\}$ is the class label (category/domain) of G_i, we learn a multiclass classification model $f : \mathbf{x} \rightarrow y$. This classification model is used to predict the category (domain) of a new arbitrary unknown network G' described only by a D-dimensional feature vector $\mathbf{x}'$. The category refers to the problem domain or area for real-world networks. For synthetic graphs a category refers to the specific graph model (*i.e.*, synthetic graph generator) used to generate a given graph. In this work, we focus on learning a classification model f using random forests (RF) [13], *i.e.*, an ensemble of decision trees where each decision tree is learned using a randomly selected subset of features. We also considered Gaussian Naïve Bayes (GNB) [19], support vector machines (SVM) [16] and logistic regression (LR) [9]. Notably, these multiclass classification models all performed very similar to random forests and therefore were removed for brevity. Random forests are favored since they performed slightly better than the other models while also being based on decision trees which are simple, efficient, and easy to interpret.

The classification model was trained using $N = 785$ networks from $K = 13$ categories (See Figure 1) which are characterized by $D = 12$ structural features. We also investigate a classification model that uses only $D = 4$ features. For evaluation, we use leave one out cross validation (LOOCV) [19]. LOOCV progressively removes each data point, trains the model then attempts to classify the removed data-point. LOOCV was selected over the more standard k-fold validation due to the small number of networks available to us in some network domains.

3 RESULTS

Prediction results: To answer the question of whether the category of an unknown network can be accurately predicted, we learn a multiclass classification model f using simple graph features and use it for prediction. The full classification results using $D = 12$ graph features are provided in Figure 1, including precision and recall for each category of networks. Notably, we achieve 96.6% accuracy in classification using a random forest model. These results support several important findings. First, we find that standard classification algorithms are able to accurately predict the category (domain) of both real-world networks and synthetically generated graphs. Second, we observe that synthetically generated graphs from 5 different synthetic graph models are easily distinguishable from real-world networks as shown in Figure 1. For instance, the synthetic graphs are distinct enough from their real-world counterparts that only seven other networks are classified as either BA, CL, ER, KPGM, or SW. Nevertheless, we are able to correctly predict that a graph is a synthetic graph 100% of the time, but more importantly, we can even predict the specific graph model that it arises from with 100% accuracy across all 5 different synthetic graph models. In other words, the synthetically generated graphs generated by the different graph models are themselves easy to distinguish between. For example, the structure of KPGM graphs are fundamentally different from CL graphs. Furthermore, we are also able to correctly classify that an arbitrary graph is not only *synthetic* or not, but also the specific graph model used to generate it. This observation indicates that synthetic graphs derived from these graph models have low variance, and thus form tightly-knit clusters that are structurally distinct from other synthetic graphs as well as real-world networks. This result is surprising since synthetic graph generators are usually evaluated based on whether they preserve the properties of real-world networks (*e.g.*, see [31]).

Careful analysis of the mislabeled graphs in Figure 1 provides interesting network/category specific findings and suggestions. For example, 10 of the 36 brain networks are non-human, and all 6 graphs that are mislabeled in Figure 1 are non-human. This is strong evidence that either the human brain networks are truly distinct from the non-human brain networks, or the network discovery process is not sufficiently standardized for brain networks. Another interesting observation is that a visual inspection of the graphs mislabeled as retweet networks show surprising similarities to one another. This suggests that in addition to predicting the domain of arbitrary networks, classification models can also provide valuable insight into alternative research techniques for crossing disciplines.

We now investigate the following question: What is the smallest most basic set of graph features that can be used to accurately

ACTUAL \ PREDICTED	Brain	Chem	Eco.	FB	RT	Road	Soc.	Web	BA	CL	ER	KPGM	SW	Recall
Brain	28	1	4	0	0	0	0	0	0	0	1	1	1	0.78
Chem	0	119	0	0	0	0	0	0	0	0	0	0	0	1
Ecology	0	0	6	0	0	0	0	0	0	0	0	0	0	1
Facebook	0	0	0	110	0	0	2	0	0	0	0	0	0	0.98
Retweet	0	0	0	0	57	0	2	0	0	1	0	0	1	0.93
Road	0	0	0	0	0	4	0	0	0	0	0	0	1	0.80
Social	0	2	0	1	1	0	35	2	0	3	0	2	0	0.76
Web	0	0	0	1	0	0	10	6	0	0	0	0	0	0.35
Barabasi	0	0	0	0	0	0	0	0	50	0	0	0	0	1
Chung-Lu	0	0	0	0	0	0	0	0	0	75	0	0	0	1
Erdős–Rényi	0	0	0	0	0	0	0	0	0	0	75	0	0	1
KPGM	0	0	0	0	0	0	0	0	0	0	0	75	0	1
Small-world	0	0	0	0	0	0	0	0	0	0	0	0	108	1
Precision	1	0.98	0.6	0.98	0.98	1	0.71	0.75	1	0.95	0.99	0.96	0.97	

Figure 2: Classification results using only 4 features. The random forest classification model is learned using 4 simple and computationally efficient features. Despite using only 4 features, we are able to accurately predict with 95.3% accuracy the category (domain) of arbitrary networks. Furthermore, the difference in classification performance compared to the previous model using 12 features is small, *e.g.*, 96.5% accuracy is achieved with 12 features compared to 95.3% accuracy using only 4 features.

predict the category (domain) of an arbitrary unknown network? The accuracy, recall and precision should be close to the previous model learned that uses the set of 12 features (Figure 2). We learn a random forest model using only density $\rho(G)$, average degree d_{avg}, assortativity $r(G)$, and maximum k-core number $K(G)$ as features. Classification results using only these four simple features are provided in Figure 2. The overall classification accuracy is 95.3%. Importantly, the results in Figure 2 indicate that a networks domain can be accurately predicted with only a few features that are all computationally efficient with a time complexity of at most $O(|E|)$. Notice the difference in accuracy, recall, and precision compared to Figure 1 is small. In this experiment, we removed the features that tend to correlate with the size of the network such as the maximum degree and total triangles. This provides additional evidence that different categories of networks from a variety of domains have distinct structural properties that can be used to learn a model to accurately distinguish between them. Observe that we are still able to correctly classify all the synthetic graphs that arise from the 5 different synthetic graph models. Furthermore, the classification recall for prediction of ecology networks actually improves using these 4 simple features.

Feature analysis: To gain further understanding of the previous classification results, we analyze the possible correlations between the features and network categories below. To understand the potential correlations between the graph features, we measure the pairwise Pearson correlation between each pair of features $\mathbf{C} = \Phi\langle \mathbf{x}_i, \mathbf{x}_j \rangle, \ \forall i, j$ where $\mathbf{C}$ is a $D \times D$ symmetric correlation matrix, Φ is a similarity function which in this case is Pearson correlation. The correlation matrix is shown in Figure 3 where 1 is a positive linear correlation, 0 is no correlation, and -1 is negative correlation. One important finding from Figure 3 is that many of the features are either not correlated at all (*i.e.*, C_{ij} is close to 0) or weakly correlated.

There are a few notable exceptions including total triangles $T(G)$ and the maximum k-core number of a graph G denoted by $K(G)$. Note that $K(G)$ provides an upper bound on the maximum clique. Meanwhile, the more triangles that exist in the graph, the more likely a larger clique is to form in G [45]. A triangle is also a 3-node clique, and thus larger cliques are obviously made up of a lot of smaller 3-node triangles. We also observe weaker correlation between a few other graph features. For instance, the maximum clique $\omega(G)$ and maximum k-core number $K(G)$ are somewhat correlated with the average degree d_{avg} and average triangles T_{avg}. In addition, the maximum k-core number $K(G)$ is somewhat correlated with the maximum number of triangles T_{max}. This suggests that these features contain potentially redundant information and are reasonable candidates for removal.

To understand the correlations between the *network categories* using these simple graph features, we use a centroid feature vector $\bar{\mathbf{x}}_i$ for each network category i. More formally, let $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots \mathbf{x}_{N_i} \end{bmatrix}$ be the graph feature matrix for network category i, and let N_i denote the number of networks in category i, then the centroid feature vector $\bar{\mathbf{x}}_i$ for network category i is:

$$\bar{\mathbf{x}}_i = \frac{1}{N_i} \sum_{j}^{N_i} \mathbf{x}_j \ = \ \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_{N_i} \tag{5}$$

where $\bar{\mathbf{x}}_i$ is a D-dimensional centroid feature vector for network category i. The above is repeated for each network category. Afterwards, we measure pairwise Pearson correlation between each pair of centroid graph feature vectors for each network category $\mathbf{C} = \Phi\langle \bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j \rangle, \ \forall i, j$ where $\mathbf{C}$ is a $K \times K$ symmetric correlation matrix and Φ is the Pearson correlation function. The network

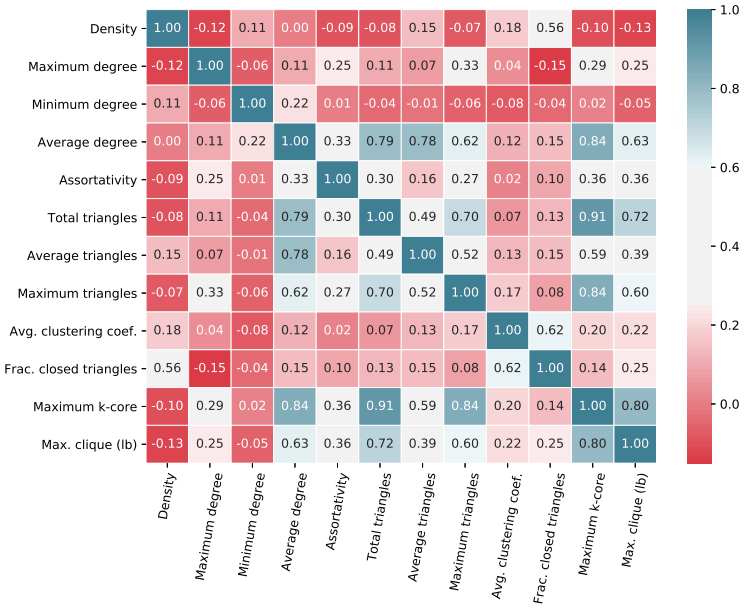


Figure 3: Graph feature correlations. To understand the correlations between the graph features, we measure pairwise Pearson correlation between each pair of features $\mathrm{C} = \Phi\langle \mathbf{x}_i, \mathbf{x}_j \rangle,\ \forall i, j$ where C is a $D \times D$ symmetric correlation matrix and Φ is the Pearson correlation function.

category correlation matrix is provided in Figure 4. There are a few interesting observations from Figure 4. Notably, we see that social networks and web graphs appear strongly correlated, which provides additional evidence for this observation mentioned earlier. We also find that cheminformatics networks appear to be correlated with ecology networks when Pearson correlation is used, which is somewhat surprising. Finally, we also notice a strong negative correlation between retweet networks and road networks.

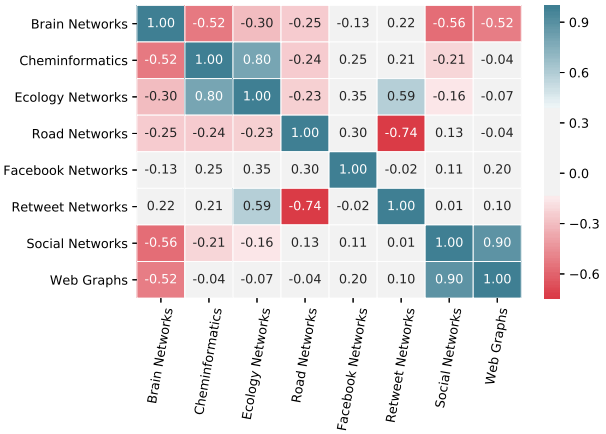


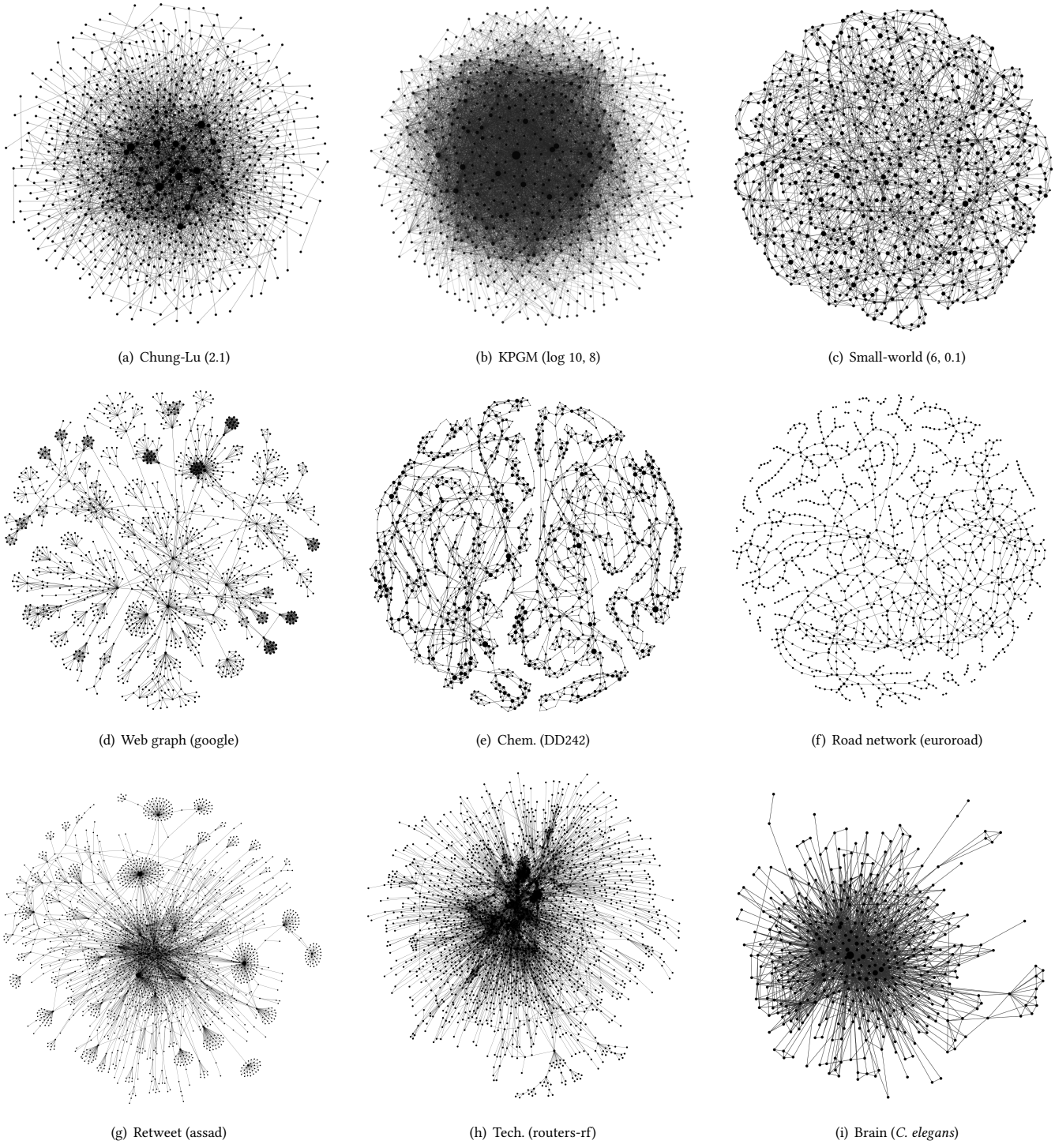
Figure 4: Network category correlations. To understand the potential structural similarities between the different network categories (domains), we derive a centroid feature vector for each network category and measure pairwise Pearson correlation.

Visual comparison of networks: Figure 5 visually compares networks from different domains and graph models using node-link visualizations. Obviously, the goal of any synthetic graph model is to generate networks that closely match the structure of real-world networks. In other words, they seek to generate realistic networks that preserve the known structural properties of networks observed in the real-world. We observe that synthetic graphs shown in Figure 5(a)-5(c) do not resemble any of the real-world networks in Figure 5(d)-5(i) from six different domains. Most of these synthetic graph generators are able to generate graphs with realistic and expected degree distributions, but fail to capture other key structural properties that are important in distinguishing the real-world networks. For instance, we observe in Figure 5(d) that web graphs have many large cliques representing a tightly-knit communities of web pages while also containing large stars that representing important hub web pages that connect the web surfer to important web pages. Other important structural patterns can be observed in the other networks from different domains (Figure 5).

4 RELATED WORK

Related research is categorized into four areas: (1) within-domain graph classification, (2) synthetic graph classification, (3) graph similarity and matching, and (4) graph feature learning and extraction methods.

Within-domain graph classification: Most previous work has focused on classification of graphs within a particular category (domain) such as molecular graphs [20, 21, 30, 33, 41, 54]. Other examples include distinguishing between brain or breast cancer cells [32] or distinguishing between different social structures [51].



(a) Chung-Lu (2.1)

(b) KPGM (log 10, 8)

(c) Small-world (6, 0.1)

(d) Web graph (google)

(e) Chem. (DD242)

(f) Road network (euroroad)

(g) Retweet (assad)

(h) Tech. (routers-rf)

(i) Brain (*C. elegans*)

Figure 5: Comparing the structure of networks from different domains and network models visually using node-link diagrams. In (a)-(c), we visualize the structure of three synthetically generated networks from different graph models whereas (d)-(i) are from different domains. Notice road networks (f) typically have long chains (paths) with small maximum degrees, and that retweet networks (g) consist of mainly large stars while web graphs (d) have large stars and large cliques.

This previous work tries to classify graphs from the same category/domain into subcategories (or types). We call this problem the *within-domain graph classification problem*. However, in this work we focus on the *across-domain graph classification problem* as well as predicting the underlying network model (generative process) used to generate a particular synthetic graph. Another key difference is that most research in this area has focused on developing more accurate and better algorithms for the within-domain graph classification problem. For instance, Gärtner *et al.* [21] proposed an approach based on graph kernels. There has been numerous other work focused on deriving new graph kernels for within-domain graph classification [20, 33, 41, 48, 54]. More recently, Shervashidze *et al.* [49] proposed more efficient graphlet kernels for within-domain graph classification.

Synthetic graph classification: Another related area of research deals with classification of synthetic graphs according to the synthetic generator that produced them [11]. However, most work in this area has simply used synthetic graphs as a way to evaluate/benchmark a proposed method. For instance, Bonner *et al.* [11] proposes a new approach called deep topology classification and evaluates the proposed method using synthetic graphs. Other work that used synthetic graphs for evaluation has mainly focused on parallel algorithms for comparing such graphs [12]. However, in this work we investigate whether we can classify synthetic graphs using standard classification models with simple graph features. In particular, we find that such graphs are trivial to classify and that synthetic graphs from a particular generator forms a tight cluster with extremely small variance, which makes these graphs trivial to classify correctly. This result is significant as it implies that using synthetic graphs for evaluation as done previously should be done with extreme caution. Moreover, this finding also highlights the limitations and problems of existing graph models and synthetic graph generation algorithms. In particular, one obvious problem is that the graphs generated from such models have extremely low variance and essentially all appear to be extremely similar. More importantly, the goal of synthetic graph models is to derive synthetic graphs that are very similar to real-world graphs (*e.g.*, for use in simulations, algorithm benchmarking, etc.) and therefore the observations made in our work highlight the inability of these models for deriving graphs that appear similar to any category of real-world networks investigated.

Graph similarity & matching: Recent research has focused on measuring the similarity between graphs from the same domain [1, 22, 42, 44, 56]. There has also been a lot of work on graph matching and network alignment [26, 27, 29, 34, 35]. Koutra *et al.* [28] proposed a fast graph alignment method for aligning large bipartite graphs. Other work has focused on fast and parallel algorithms for the matching problem [27] as well as parallel approximation algorithms for network alignment [26]. There has also been a lot of work on graph matching using graphlet and network motif features [29, 34, 35]. More recently, Soundarajan *et al.* [50] reviewed many different graph similarity measures for comparing graphs. Other work by Ali *et al.* [7] has focused on sub-sampling techniques for network comparison whereas Onnela *et al.* [40] presented a taxonomy of networks based on community structures. However, all of this work focuses on fundamentally different problems. In contrast, this paper investigates whether or not the domain (category) of an arbitrary network can be predicted accurately. Recent work by Ikehara [25] analyzed networks from different domains and found some to be difficult to distinguish based on structure alone. This observation is in contrast to our own work, which demonstrates that the domain of most networks can be accurately predicted with 96.6% accuracy. More importantly, that work mainly focused on understanding and *analyzing* the differences between networks from different domains, while our goal is to study whether the domain (category) of a network can be *predicted* using a multiclass classification model with simple graph features. While Ikehara [25] analyzes networks using *complex graphlet features* [3], our work shows that *simple graph features* are sufficient to *predict* the category of networks. Additionally, Ikehara [25] studies the *binary classification problem* between 6 different network categories, whereas we focus on the more important and challenging *multiclass classification problem* between 13 different categories of networks. As an aside, since Ikehara [25] focused on binary classification they had to carefully handle the class imbalance problem, whereas in this work it is less of an issue since we focus on multiclass classification [8, 53].[2]

Graph feature learning & extraction: There are also many methods for automatically learning a graph feature representation [5, 17, 30, 48]. Most methods are not inductive and explicitly assume that the graphs are from the same domain and the node identifiers used in the various graphs are consistent. More recently, inductive methods for learning graph feature representations have been proposed [47]. These methods allow the learned features to be transferred across networks and thus can be used for classifying graphs from different domains (which is the problem investigated in this work).

5 CONCLUSION

In this work, we investigated whether the domain or generative process of a complex network can be accurately predicted using only a handful of simple graph features. Our results indicate that networks drawn from different domains (and network models) are trivial to distinguish using only a few graph features. In particular, we achieve 96.6% accuracy using a simple random forest model to predict the domain and/or generative process governing the formation of the network. The full classification results are provided in Figure 1, including precision and recall for each category of networks. We also achieved an accuracy of 95.3% when using a reduced feature vector of only 4 features, with similar details provided in Figure 2. This implies that real-world complex networks from various domains have distinct structural properties (acting as a signature) that allow us to predict with high accuracy the domain (category) of an arbitrary network. Furthermore, synthetic graphs are trivial to classify as the model can predict with near-certainty whether a

[2] For instance, in binary classification, if the majority of points are of one class, then one can achieve a reasonable accuracy by simply predicting all points to be of the majority class. However, in multiclass classification, if we were to predict the majority class for each point, then we would achieve very poor accuracy since we predict the class (category) of around 800 graphs, and the largest class in our data consists of only about 100 points, and therefore we would achieve an accuracy of less than 20% if the classification model simply predicted all points as the majority class.

graph is synthetic or not but more importantly the network model used to generate the synthetic graph.

Extension of work: The results and findings in this work also have a variety of practical applications beyond expanding our understanding of real-world complex networks and synthetic graph models. In particular, the models learned in this work to accurately predict the category (domain) of a network can be used in network data repositories (data archives) such as NetworkRepository [43]. For instance, suppose a user donates an arbitrary network, we can then use the multiclass classification models to recommend a category (domain) and possibly other metadata that was not provided by the user. In addition, we can use the results of this work to recommend "structurally related networks" to users. For instance, if a user is analyzing a particular network using the interactive visual graph mining tools provided by NR [43], then we can automatically recommend other relevant graphs that are structurally similar to the network being analyzed by the user.

Furthermore, we are also currently using the key findings of this work to build a "graph search engine." The engine would allow users to search for graphs that are structurally similar to the graph of interest given as input by the user. In particular, given a graph G provided as input by a user, we compute a few fast and efficient structural properties from G denoted by $\mathbf{y}$. We then derive

$$\mathbf{r} = \mathcal{K}(\mathbf{y}, \mathbf{x}_i), \quad \text{for } i = 1, 2, \ldots, |\mathcal{G}| \tag{6}$$

where $\mathcal{G}$ is the set of graphs in our graph database (*e.g.*, all graphs available at NR [43]); $\mathcal{K}$ is a similarity function between the input graph G and each graph $G_i \in \mathcal{G}$; and $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \cdots \end{bmatrix}$ is a score vector. Each r_i indicates how similar G is to $G_i \in \mathcal{G}$. Thus, we can simply sort $\mathbf{r}$ and return the top-k graphs that closely resemble the input graph G provided by the user.

ACKNOWLEDGMENTS

We thank all the reviewers for many helpful suggestions and feedback. This research was supported by a National Science Foundation (NSF) REU grant.

REFERENCES

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 624–632. ACM, 2012.
[2] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield. Efficient graphlet counting for large networks. In *International Conference on Data Mining*, pages 1–10. IEEE, 2015.
[3] N. K. Ahmed, J. Neville, R. A. Rossi, N. Duffield, and T. L. Willke. Graphlet decomposition: Framework, algorithms, and applications. *Knowledge and Information Systems (KAIS)*, pages 1–32, 2016.
[4] N. K. Ahmed and R. A. Rossi. Interactive visual graph analytics on the web. In *International AAAI Conference on Web and Social Media (ICWSM)*, pages 566–569, 2015.
[5] N. K. Ahmed, R. A. Rossi, R. Zhou, J. B. Lee, X. Kong, T. L. Willke, and H. Eldardiry. Representation learning in large attributed graphs. In *WiML NIPS*, 2017.
[6] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys*, 74:47, 2002.
[7] W. Ali, A. E. Wegner, R. E. Gaunt, C. M. Deane, and G. Reinert. Comparison of large networks with sub-sampling strategies. *Scientific reports*, 6:28955, 2016.
[8] M. Aly. Survey on multiclass classification methods. *Neural Networks*, 19:1–9, 2005.
[9] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
[10] S. Bonner, J. Brennan, I. Kureshi, G. Theodoropoulos, and A. McGough. Efficient comparison of massive graphs through the use of graph fingerprints. In *KDD MLG Workshop*, 2016.
[11] S. Bonner, J. Brennan, G. Theodoropoulos, I. Kureshi, and A. S. McGough. Deep topology classification: A new approach for massive graph classification. In *International Conference on BigData*, pages 3290–3297. IEEE, 2016.
[12] S. Bonner, J. Brennan, G. Theodoropoulos, I. Kureshi, and A. S. McGough. GFP-X: A parallel approach to massive graph comparison using spark. In *International Conference on Big Data*, pages 3298–3307. IEEE, 2016.
[13] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
[14] J. P. Canning, E. E. Ingram, S. Nowak-Wolff, A. M. Ortiz, N. K. Ahmed, R. A. Rossi, K. R. B. Schmitt, and S. Soundarajan. Network classification and categorization. In *arXiv:1709.04481*, September 2017.
[15] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.
[16] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
[17] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pages 2224–2232, 2015.
[18] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.
[19] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
[20] T. Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):49–58, 2003.
[21] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines*, pages 129–143, 2003.
[22] T. E. Goldsmith and D. M. Davenport. Assessing structural similarity of graphs. 1990.
[23] Graph500. http://graph500.org/?page_id=12.
[24] T. Guo and X. Zhu. Understanding the roles of sub-graph features for graph classification: an empirical study perspective. In *CIKM*, pages 817–822. ACM, 2013.
[25] K. Ikehara. *The Structure of Complex Networks across Domains*. PhD thesis, University of Colorado at Boulder, 2016.
[26] A. M. Khan, D. F. Gleich, A. Pothen, and M. Halappanavar. A multithreaded algorithm for network alignment via approximate matching. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 64. IEEE Computer Society Press, 2012.
[27] G. Kollias, M. Sathe, O. Schenk, and A. Grama. Fast parallel algorithms for graph similarity and matching. *Journal of Parallel and Distributed Computing*, 74(5):2400–2410, 2014.
[28] D. Koutra, H. Tong, and D. Lubensky. Big-align: Fast bipartite graph alignment. In *International Conference on Data Mining*, pages 389–398. IEEE, 2013.
[29] N. Kriege and P. Mutzel. Subgraph matching kernels for attributed graphs. *arXiv preprint arXiv:1206.6483*, 2012.
[30] J. B. Lee, R. Rossi, and X. Kong. Deep graph attention model. In *arXiv:1709.06075*, 2017.
[31] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *JMLR*, 11(Feb):985–1042, 2010.
[32] G. Li, M. Semerci, B. Yener, and M. J. Zaki. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):265–283, 2012.
[33] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Extensions of marginalized graph kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 70. ACM, 2004.
[34] N. Malod-Dognin and N. Pržulj. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13):2182–2189, 2015.
[35] T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer informatics*, 9:121, 2010.
[36] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
[37] M. Newman. *Networks: an introduction*. Oxford university press, 2010.
[38] M. E. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.
[39] M. E. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.
[40] J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, M. D. Fricker, and N. S. Jones. Taxonomies of networks from community structure. *Physical Review E*, 86(3):036104, 2012.
[41] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110, 2005.
[42] J. W. Raymond, E. J. Gardiner, and P. Willett. Rascal: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.
[43] R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[44] R. A. Rossi and N. K. Ahmed. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1112–1131, 2015.

[45] R. A. Rossi, D. F. Gleich, and A. H. Gebremedhin. Parallel maximum clique algorithms with applications to network analysis. *Journal on Scientific Computing (SISC)*, 37(5):28, 2015.

[46] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and M. M. A. Patwary. Fast maximum clique algorithms for large graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 365–366. ACM, 2014.

[47] R. A. Rossi, R. Zhou, and N. K. Ahmed. Deep feature learning for graphs. In *arXiv:1704.08829*, pages 1–11, 2017.

[48] N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[49] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pages 488–495, 2009.

[50] S. Soundarajan, T. Eliassi-Rad, and B. Gallagher. A guide to selecting a network similarity method. In *SIAM International Conference on Data Mining*, pages 1037–1045. SIAM, 2014.

[51] J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1307–1318. ACM, 2013.

[52] M. van Steen. *Graph Theory and Complex Networks*. Maarten van Steen, first edition, apr 2010.

[53] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[54] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt. Graph kernels. *JMLR*, 11:1201–1242, 2010.

[55] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[56] L. A. Zager and G. C. Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.