Social Network Analysis and Mining (2019) 9:51
https://doi.org/10.1007/s13278-019-0593-7

ORIGINAL ARTICLE

Complex networks are structurally distinguishable by domain

Ryan A. Rossi[1] · Nesreen K. Ahmed[2]

Received: 19 March 2019 / Revised: 20 August 2019 / Accepted: 27 August 2019 / Published online: 19 September 2019

Abstract

Complex networks arise in many domains and often represent phenomena such as brain activity, social relationships, molecular interactions, hyperlinks, and re-tweets. In this work, we study the problem of predicting the category (domain) of arbitrary networks. This includes complex networks from different domains as well as synthetically generated graphs from six different network models. We formulate this problem as a multiclass classification problem and learn a model to predict the domain of a new previously unseen network using only a small set of simple structural features. The model is able to accurately predict the domain of arbitrary networks from 17 different domains with 95.7% accuracy. 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 graph model used to generate it. Overall, the results demonstrate that networks drawn from different domains and graph models are distinguishable using a few simple structural features.

Keywords Network categorization · Structural properties · Network science · Complex networks · Network classification

1 Introduction

Networks that arise in different domains often represent phenomena such as molecular interactions, hyperlinks, re-tweets, and brain activity. While there are inherent similarities in network structure across different domains (e.g., a power-law degree distribution), it is also generally believed that networks from different domains 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 95.7% accuracy) the category of a new arbitrary network G' described only by a D-dimensional feature vector $\mathbf{x}'$, where $y \in \{1, 2, \dots, K\}$ 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 1013 networks from $K = 17$ categories (see Fig. 1) that are characterized using 11 simple graph features. Studying this problem allows us to gain further understanding of complex networks across different domains and the different underlying phenomena that govern the formation and structure of such complex networks.

We also investigate a classification model that uses only four features for predicting the category of unknown networks. In particular, the four simple features used are density, average degree, assortativity, and maximum k-core. These structural features were selected since they are computationally efficient 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) (Milo et al. 2002; Ahmed et al. 2015) are likely to further improve the accuracy. However, such complex structural features are more computationally expensive, 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, the results show that networks from different domains can be accurately distinguished using only the most basic and fundamental structural properties.

✉ Ryan A. Rossi
rrossi@adobe.com

Nesreen K. Ahmed
nesreen.k.ahmed@intel.com

[1] Adobe Research, 345 Park Ave, San Jose, CA, USA

[2] Intel Labs, 3065 Bowers Ave, Santa Clara, CA, USA

		PREDICTED																
		Brain	Chem	Eco.	Econ.	FB	Pow.	RT	Road	SC	Soc.	Web	BA	CL	ER	KPGM	PLC	SW	Recall
ACTUAL	Brain	109	1	3	1	0	0	2	0	0	0	0	0	0	0	1	0	0	0.93
	Chem	0	119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Ecology	1	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.83
	Economic	0	0	0	13	0	0	2	0	0	0	0	0	1	0	0	0	0	0.81
	Facebook	0	0	0	0	111	0	0	0	0	1	0	0	0	0	0	0	0	0.99
	Power	0	0	0	0	0	5	0	1	0	0	1	0	0	0	0	0	1	0.63
	Retweet	0	0	0	0	0	1	57	0	0	3	0	0	0	0	0	0	0	0.93
	Road	0	0	0	0	0	0	0	13	0	1	0	0	0	0	0	0	1	0.87
	Sci. Comp.	0	0	0	0	0	0	1	0	8	1	0	0	0	0	0	0	1	0.73
	Social	0	1	0	0	1	0	1	0	0	37	2	0	2	0	1	0	1	0.80
	Web	1	0	0	0	0	0	0	0	0	8	10	0	0	0	0	0	0	0.53
	Barabasi	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	1
	Chung-Lu	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	1
	Erdős–Rényi	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	1
	KPGM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	1
	PLC	0	0	0	0	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	0	0	0	0	108	1
	Precision	0.98	0.98	0.63	0.93	0.99	0.83	0.90	0.93	1	0.73	0.77	1	0.96	1	0.97	1	0.96	95.76%

Fig. 1 Prediction results using 11 features. The model correctly predicts the category/domain of the networks with an accuracy of 95.7%. These results indicate that networks from different domains/categories are structurally distinguishable

Previous research has mainly focused on either (1) classification of synthetic graphs (Bonner et al. 2016b) or (2) graphs within a particular category/domain such as molecular graphs (Vishwanathan et al. 2010; Ralaivola et al. 2005; Lee et al. 2017). Other examples include distinguishing between brain or breast cancer cells (Li et al. 2012) or distinguishing between different social structures (Ugander et al. 2013). 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 has used synthetically generated graphs (Bonner et al. 2016a), 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 (Guo and Zhu 2013; Li et al. 2012). In those domains, generating a large number of graphs from the similar phenomenon is still relatively simple.

Our contributions. To improve our understanding of complex networks, 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 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 95.7% 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. While the main motivation for studying this problem is to improve our understanding of complex networks (and synthetic graph models), the results and findings can also be used in many other ways, e.g., to recommend (or find) networks that are structurally similar to an unknown network given as input by the user (graph search engine), or to build better synthetic graph generators and improve evaluation of them. Other applications are discussed later in Sects. 3.1 and 5.

This work makes two important findings. First, real-world networks from various domains have distinct structural properties that allow us to predict with high accuracy the category of an arbitrary network. Second, 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 Related work

The majority of previous research has focused on classification of graphs within a particular category (domain) such as molecular graphs (Gärtner et al. 2003; Gärtner 2003; Vishwanathan et al. 2010; Mahé et al. 2004; Ralaivola et al. 2005; Lee et al. 2017). Other examples include distinguishing between brain or breast cancer cells (Li et al. 2012) or distinguishing between different social structures (Ugander et al. 2013). We call this problem the *within-domain graph classification problem*. This problem is fundamentally different from the one investigated in this work. In contrast, our work focuses on the *across-domain graph classification problem* as well as predicting the underlying network model (generative process) used to generate a particular synthetic graph.

For the within-domain graph classification problem, most research has focused on developing more accurate and better algorithms. For instance, Gärtner et al. (2003) proposed an approach based on graph kernels. There has been numerous other work focused on deriving new graph kernels for within-domain graph classification (Gärtner 2003; Mahé et al. 2004; Ralaivola et al. 2005; Vishwanathan et al. 2010; Shervashidze et al. 2011). Alternatively, Shervashidze et al. (2009) proposed more efficient graphlet kernels for within-domain graph classification. In contrast, our work does not focus on developing new algorithms for classification. Instead, we leverage existing classification methods to answer two main questions: (1) Are network categories distinguishable from one another? (2) and what is the minimum set of features required to accurately predict the categories of arbitrary networks?

Classification of synthetic graphs according to the generator that produced them is another related research problem. However, most work has simply used synthetic graphs as a way to evaluate/benchmark a proposed method. For instance, Bonner et al. (2016b) proposed a new approach called deep topology classification and evaluated the proposed method using synthetic graphs. Other work that used synthetic graphs for evaluation has mainly focused on parallel algorithms for comparing such graphs (Bonner et al. 2016c). 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 this work highlight the inability of these models for deriving graphs that appear similar to real-world networks.

Research focused on measuring the similarity between graphs from the same domain has also received considerable attention (Goldsmith and Davenport 1990; Raymond et al. 2002; Zager and Verghese 2008; Abrahao et al. 2012; Rossi and Ahmed 2015b). There has also been a lot of work on graph matching and network alignment (Khan et al. 2012; Milenković et al. 2010; Kriege and Mutzel 2012; Kollias et al. 2014; Malod-Dognin and Pržulj 2015). Koutra et al. (2013) proposed a fast graph alignment method for aligning large bipartite graphs. Other work has focused on fast and parallel algorithms for the matching problem (Kollias et al. 2014) as well as parallel approximation algorithms for network alignment (Khan et al. 2012). There has also been a lot of work on graph matching using graphlet and network motif features (Milenković et al. 2010; Kriege and Mutzel 2012; Malod-Dognin and Pržulj 2015). More recently, Soundarajan et al. (2014) reviewed many different graph similarity measures for comparing graphs. Other work by Ali et al. (2016) has focused on sub-sampling techniques for network comparison, whereas Onnela et al. (2012) 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.

Closest in spirit to our research is work by Ikehara (2016); Ikehara and Clauset (2017), which appeared publicly at roughly the same time as our earlier work in Canning et al. (2017, 2018). However, there are a number of key differences. First, 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. Second, while Ikehara (2016) used *complex graphlet features* (Ahmed et al. 2016), our work shows that *simple graph features* are sufficient to *predict* the category of networks. Third, Ikehara (2016) studied a *binary classification problem*, whereas we focus on the *multiclass classification problem*. Finally, we also examine a different set of network categories including 11 real-world network domains and six synthetic graph models.

There are also many methods for automatically learning a graph feature representation (Shervashidze et al. 2011; Duvenaud et al. 2015; Ahmed et al. 2017; Lee et al. 2017). Most approaches 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 (Rossi et al. 2017). The features learned from these methods can be transferred across networks and therefore can be used for classifying graphs from different domains (which is the problem investigated in this work).

3 Methodology

This section formally presents the problem (Sect. 3.1) and describes the network data and categories (Sect. 3.2), the synthetic graph models and parameters (Sect. 3.3), the simple structural features used to characterize the networks (Sect. 3.4), and the classification models and techniques used for predicting the domain of arbitrary networks (Sect. 3.5).

3.1 Problem formulation

To improve our understanding of complex networks (and synthetic graph models/generators), we investigate the following problem:

Problem 1 *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* $y_i \in \{1, 2, \dots, K\}$ *is the class label (category/domain) of* G_i, *we learn a multiclass classification model* $f : \mathbf{x} \rightarrow y$. *The classification model f is then used to predict the category (domain)* y' *of a new arbitrary unknown network* G' *described only by a D-dimensional feature vector* $\mathbf{x}'$. *More formally, given f and the D-dimensional feature vector* $\mathbf{x}'$ *for* G', *the category* y' *is predicted as*

$$\hat{y}' = f(\mathbf{x}'), \tag{1}$$

where $\hat{y}'$ *is the predicted category of* G' *and* y' *is the actual ground-truth category.*

The above problem has two main parts: (1) learning a model *f* from training data and then (2) using the model to infer the domain/category of unknown networks. As an aside, the category refers to the domain for real-world networks. For synthetic graphs, a category refers to the specific graph model (synthetic graph generator) used to generate a given graph.

While the main motivation for studying this problem is to *improve our understanding of complex networks* and synthetic graph models, the results and findings of this work can also be used for many other important applications:

- Find networks and categories that are structurally similar to a graph of interest (given as input by the user).
- Use results to improve the metadata in data repositories, e.g., suppose a user donates an arbitrary network, then we can use the model to recommend a category/domain and possibly other metadata based on the structural properties alone.
- Improve evaluation and understanding of synthetic graph models and generators.

Many other potential applications of this work are discussed in Sect. 5.

3.2 Network data and categories

In this work, we use a data set consisting of 1013 graphs from $K = 17$ categories/domains. The network classification data set includes real-world networks from 11 different domains/categories along with synthetic graphs from six different graph models. A list of the network categories is shown in Fig. 1.

3.2.1 Real-world network data

Data were obtained from the Network Repository (NR) (Rossi and Ahmed 2015a) for all non-synthetic graphs.[1] We accessed the data from NR on May 25, 2017. This included complex networks from 11 different domains/categories. The categories (class labels for the graphs) are naturally defined based on the underlying domain of the complex network data. This work uses the categories/domains from NetworkRepository (last accessed May 25, 2017). A list of the network categories is provided in Fig. 1 (first 11 rows/columns). NR also includes collections of networks from computational and algorithmic challenges (DIMACS, DIMACS10 and BHOSLIB), dynamic networks, temporal reachability graphs, and a few others. As these collections are fundamentally different from static networks from a discipline or field, they were discarded as outside the problem scope. Finally, the cheminformatics category had significantly more instances than all others and therefore was downsampled to be comparable to the next largest category.

3.2.2 Synthetic graph data

In addition to this large collection of real-world networks, we also generated synthetic graphs from six different graph models including: 75 from Barabási–Albert (BA) graph model (Albert and Barabási 2002), 75 from Chung-Lu (CL) graph model (Chung and Lu 2002), 75 from Erdős–Rényi (ER) model (Erdős and Rényi 1960), 75 from Kronecker Product Graph Model (KPGM) (Leskovec et al. 2010), 75 from Power-Law Clustering (PLC) graph model (Holme and Kim 2002), and 108 from Watts–Strogatz Small-World (SW) model (Watts and Strogatz 1998). These six different graph models and the parameters used to generate graphs from each of them are described in Sect. 3.3. The final data set consists of 1013 graphs from $K = 17$ categories.

[1] http://networkrepository.com.

3.2.3 Data availability

In the spirit of reproducible research and future work on this problem, the network classification data used in this study have been made available online: http://networkrepository.com/classification/data.csv

It can also be explored in real time over the Web using an interactive visual graph analytics tool (Ahmed and Rossi 2015). This tool can be accessed at: http://networkrepository.com/classification

The structure including the node and links of individual networks and their properties can also be interactively visualized and explored online at: http://networkrepository.com/graphvis

3.3 Synthetic graph models and settings

This section describes the six different synthetic graph generators used in this work along with the graph model parameters used for each graph generator. For these generators, we always ensure the graphs returned have variance. For instance, we select n to be approximately around the specific n by selecting a random number that is within $+/-20\%$ of n (i.e., $+/-20\%$ of 1000, 10,000, and 100,000).

3.3.1 Barabási–Albert (BA) graph model

The Barabási–Albert (BA) preferential attachment model (Albert and Barabási 2002) 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 $p_i = \frac{k_i}{\sum_j k_j}$, 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 1000 nodes using $\sigma \in \{10, 40, 60\}$, 25 BA graphs with 10,000 nodes using $\sigma \in \{40, 60, 100\}$, and 25 BA graphs with 100,000 nodes using $\sigma \in \{40, 60, 100\}$.

3.3.2 Chung-Lu (CL) graph model

The Chung-Lu (CL) graph model (Chung and Lu 2002) generates a synthetic graph with a given expected degree sequence. 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

$$p_{ij} = \frac{w_i w_j}{\sum_k w_k} \tag{2}$$

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.

3.3.3 Erdős–Rényi (ER) graph model

Let $\text{ER}(n, p)$ denote an Erdős–Rényi (ER) (Erdős and Rényi 1960) 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 1000 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 $p \in \{0.0005, 0.00005, 0.000005\}$.

3.3.4 Kronecker product graph model (KPGM)

For the Kronecker product graph model (KPGM) (Leskovec et al. 2010), we follow the same methodology as described in [24]. In particular, the initiator matrix used is: $\begin{bmatrix} 0.57 & 0.19 \\ 0.19 & 0.05 \end{bmatrix}$. The number of nodes is $n = 2^k$, where $k \in \{8, 10, 12, 14, 16\}$ and the number of edges is αn, 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.

3.3.5 Power-law clustering (PLC) graph model

We also use the power-law clustering (PLC) graph model (Holme and Kim 2002) for generating synthetic graphs with power-law degree distributions and approximate average clustering. This model is similar to the BA graph model with an additional step that adds an edge to close a triangle with some probability p. We generate 25 PLC graphs for each $n \in \{1000, 10,000, 100,000\}$. There are two other parameters: (1) the number of edges σ to create between a new node and the existing nodes and (2) the triad closure probability p given to each random edge such that it has a chance of creating an edge to one of its neighbors too and therefore closing a triangle. Given n, the other parameters σ and p are selected uniformly at random from $\sigma \in \{10, 40, 60\}$ and $p \in \{0.3, 0.2, 0.1\}$.

3.3.6 Small-world (SW) graph model

We also use synthetic graphs generated by the Watts–Strogatz small-world graph model (Watts and Strogatz 1998). 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 $n \in \{100, 1000, 10,000\}$, $k \in \{3, 4, 5, 6\}$, 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.

3.4 Structural graph features

In this work, we are interested in finding the simplest and most computationally efficient structural features that allow us to predict with high accuracy the domain (category) of each network data set. We represent each graph using only D simple structural features. The features used for classification are defined as follows. Although one could use more complex features, such as 4-node graphlet features (Ahmed et al. 2015), 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.

3.4.1 Normalization

All graph features are normalized appropriately using min–max scaling before using them to train the classification models. More formally, each N-dimensional feature vector $\mathbf{x}$ (e.g., average degree) is scaled as follows:

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

where $0 \leq \hat{x}_i \leq 1$, for $i = 1, \ldots, N$. This ensures the feature values in $\hat{\mathbf{x}} \in \mathbb{R}^N$ are between zero and one.

3.4.2 Feature definitions

The definitions of the features used in this work are provided as follows (van Steen 2010; Newman 2010). 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 and $d_i = |\Gamma_i|$ is the degree of i.

F$_1$ **Average degree** The average degree over all nodes in a graph G is defined as $d_{\text{avg}} = {}^1\!/_{|V|} \sum_i d_i$, where $d_i = |\Gamma_i|$ is the degree of node i.

F$_2$ **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 (Newman 2002). More formally, the assortativity coefficient of a graph G is defined as

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

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

F$_3$ **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 $K(G)$.

F$_4$ **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.

3.4.3 Other structural features

Many of the results in this work use only the four simple structural graph features defined above. However, we also investigated a model with seven additional structural features including:

F$_5$ **Maximum degree** The max degree is defined as $\Delta(G) = \max\{d_1, \ldots, d_{|V|}\}$, 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).

F$_6$ **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.

F$_7$ **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 $T(G) = \frac{1}{3} \sum_{e=(i,j)\in E} |\Gamma_i \cap \Gamma_j|$.

F$_8$ **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 $T_{\text{avg}} = {}^1\!/_{|E|} \sum_{e\in E} T_e$.

F$_9$ **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 $e = (i,j) \in E$.

F_{10} **Average clustering coefficient** The clustering coefficient of a graph quantifies how a node in a graph tends to cluster together (Watts and Strogatz 1998). 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 $C(G) = \frac{1}{N}\sum_{i \in V} C_i$.

F_{11} **Fraction of closed triangles (global clustering coefficient)** (Newman et al. 2001): 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 $\kappa(G) = T(G)/W(G)$.

Results that use "all features" leverage the seven structural features defined above as well as the initial 4 for a total of 11 graph features altogether. The different classification models along with the different feature sets used are described in Sect. 3.5.

3.5 Predictive models

In random forests, we learn each decision tree by sampling with replacement (bootstrap sampling) from the training set. Furthermore, whenever splitting a node in the tree, we select the best split among a random subset of the D features. In this work, we learn an ensemble of 50 decision trees and use entropy (information gain) to measure the quality of a split. Instead of allowing each classifier to vote for a single class as proposed originally in Breiman (2001), we combine the classifiers by averaging their probabilistic prediction. Thus, we predict the class label of an unseen graph by taking the class with largest average probability. Besides random forests, we also investigated Gaussian Naïve Bayes (GNB) (Friedman et al. 2001), support vector machines (SVM) (Cortes and Vapnik 1995) and logistic regression (LR) (Bishop 2006). However, these 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 and computationally efficient.

The classification model f is trained using N networks from the $K = 17$ categories (see Fig. 1) which are characterized by D simple structural features. The model f is then used to predict the domain of a held-out network characterized only by a D-dimensional structural feature vector $\mathbf{x}'$. More formally, given f and the structural feature vector $\mathbf{x}'$ from a new previously unseen network G', the domain/category is predicted as $\hat{y}' = f(\mathbf{x}')$, where $\hat{y}' \in \{1, \ldots, K\}$ is the predicted category. This is repeated for all networks, i.e., we hold out each network, learn a model with the others, and then predict the held-out network. This process is called leave one out cross-validation (LOOCV) (Friedman et al. 2001) and has many advantages over traditional k-fold cross-validation (CV). LOOCV is used in this work for two main reasons. First, it allows us to include network domains where there are only a small number of networks available to us. Second, traditional k-fold CV is known to have larger variance and bias than LOOCV, and thus allows us to obtain more scientifically accurate findings to the questions investigated. In general, LOOCV is typically preferred over k-fold CV as long as the computational cost involved in LOOCV is not an issue. In this work, we obviously prefer more scientifically accurate results over a slightly more convenient evaluation that is less computationally expensive.

3.5.1 Classification with different features

We investigate three different classification models that differ only in the set of structural features used for prediction. The classification models used in this work are as follows:

- **3 Features** ($\mathbf{F_1}$–$\mathbf{F_3}$): This model uses only three simple structural features to characterize each network, namely average degree, assortativity, and maximum k-core.
- **4 Features** ($\mathbf{F_1}$–$\mathbf{F_4}$): In addition to average degree, assortativity, and maximum k-core ($\mathbf{F_1}$–$\mathbf{F_3}$), this model also includes density ($\mathbf{F_4}$).
- **All Features** ($\mathbf{F_1}$–$\mathbf{F_{11}}$): This model uses 11 features to describe each network.

The 3- and 4-feature models described above use extremely simple structural features that are computationally efficient with a time complexity that is no larger than $\mathcal{O}(|E|)$. In addition to these extremely simple structural features, the model that uses $D = 11$ features also includes a few simple triangle-based features.

4 Results

The experiments are designed to answer the following important fundamental questions:

Q1 Can we correctly predict the domain/category that an arbitrary network belongs using a few simple structural properties?

Q2 Are synthetic graphs from a wide variety of graph generators difficult to distinguish from real-world complex networks? or can we accurately predict not only that a synthetic graph is indeed generated by a model, but can we also predict the synthetic graph model that an arbitrary synthetic graph was derived from?

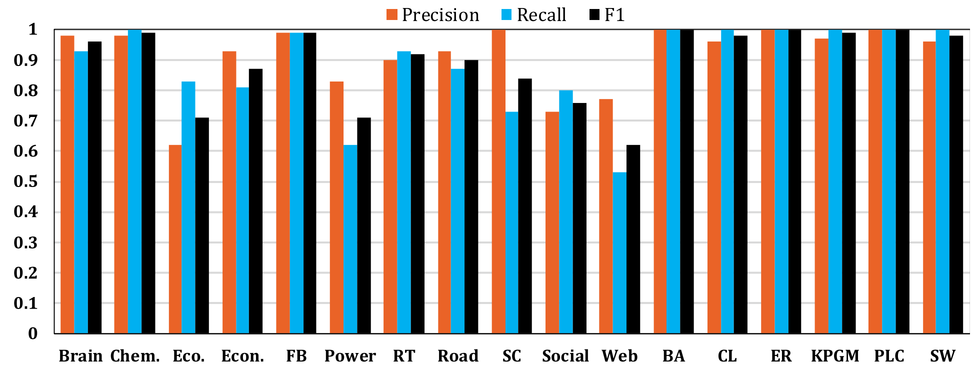


Fig. 2 Precision, recall, and F1 results for different categories of networks (using all 11 features)

Q3 What is the minimum and simplest set of features that can be used to accurately predict the category/domain of networks?

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 all 11 structural graph features are provided in Fig. 1. Notably, we achieve 95.7% accuracy in classification using a random forest model. This supports several important findings.

Result 1 *The network category (domain) of both real-world networks and synthetically generated graphs can be accurately predicted* (Fig. 1).

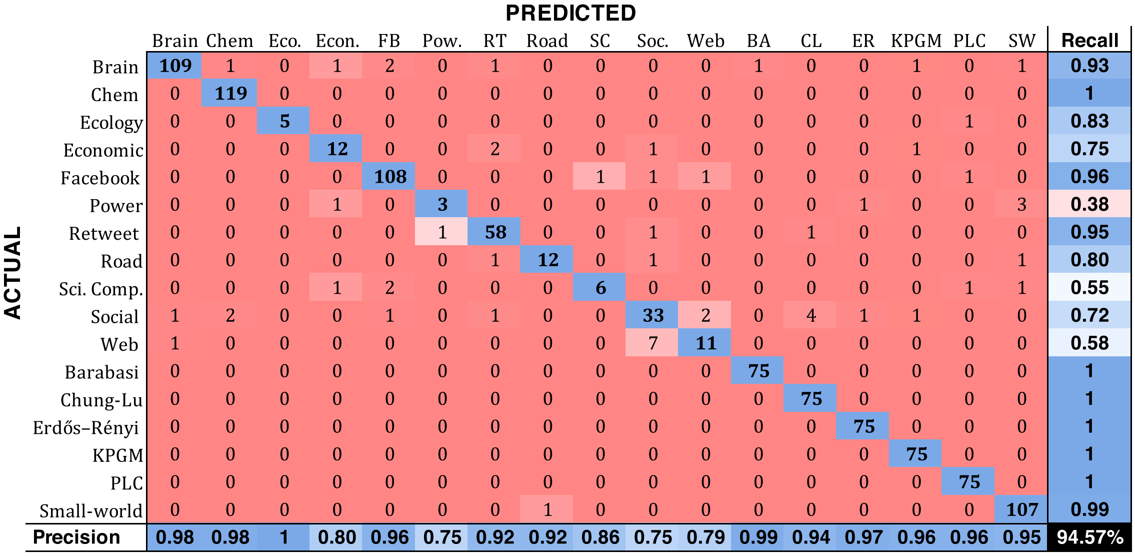
Figure 1 shows the classification results including the precision and recall for each category of networks. The model f learned using only a few simple structural features is able to accurately distinguish between graphs from different domains. An overall accuracy of 95.7% is achieved using a random forest model. In other words, the model f learned using only $D = 11$ structural features is able to predict the domain of 95.7% of the more than 1000 networks from 17 different categories/domains. Furthermore, the overall F1 score is 96%. Result 1 implies that 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. This result is important, as it not only improves our understanding of such complex networks and synthetic graph models, but also has many important high-impact applications. See Sect. 5 for discussion of a few such applications. The precision, recall, and F1 scores for the different categories of networks are shown in Fig. 2.

Network/category-specific findings and insights can also be found by analyzing the mislabeled graphs in Fig. 1. As an example, 10 of the 117 brain networks are non-human, and all 8 mislabeled graphs in Fig. 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 shows surprising similarities to one another. This suggests that in addition to predicting the domain of arbitrary networks, classification models can also provide valuable insights.

Result 2 Synthetic graphs are easily distinguishable from real-world networks as shown in Fig. 1. Moreover, the graph model used to generate a synthetic graph is trivial to predict.

Figure 1 shows that synthetically generated graphs from six different synthetic graph models are easily distinguishable from real-world networks. Synthetic graphs are distinct enough from their real-world counterparts that only nine other networks are classified as either BA, CL, ER, KPGM, PLC 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 six 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 is 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 many synthetic graph generators are evaluated based on whether they preserve the properties of real-world networks (Leskovec et al. 2010; Rossi et al. 2013; Mahadevan et al. 2007).

To further understand the significance of this finding, we also investigated a binary classification task that predicts whether a graph is synthetically generated or not. In this classification task, synthetic graphs from any of the six graph models are relabeled as “synthetic,” whereas the



PREDICTED

ACTUAL

	Brain	Chem	Eco.	Econ.	FB	Pow.	RT	Road	SC	Soc.	Web	BA	CL	ER	KPGM	PLC	SW	Recall
Brain	109	1	0	1	2	0	1	0	0	0	0	1	0	0	1	0	1	0.93
Chem	0	119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Ecology	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0.83
Economic	0	0	0	12	0	0	2	0	0	1	0	0	0	0	1	0	0	0.75
Facebook	0	0	0	0	108	0	0	0	1	1	1	0	0	0	0	1	0	0.96
Power	0	0	0	1	0	3	0	0	0	0	0	0	0	1	0	0	3	0.38
Retweet	0	0	0	0	0	1	58	0	0	1	0	0	1	0	0	0	0	0.95
Road	0	0	0	0	0	0	1	12	0	1	0	0	0	0	0	0	1	0.80
Sci. Comp.	0	0	0	1	2	0	0	0	6	0	0	0	0	0	0	1	1	0.55
Social	1	2	0	0	1	0	1	0	0	33	2	0	4	1	1	0	0	0.72
Web	1	0	0	0	0	0	0	0	0	7	11	0	0	0	0	0	0	0.58
Barabasi	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	1
Chung-Lu	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	1
Erdős–Rényi	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	1
KPGM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	1
PLC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	1
Small-world	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	107	0.99
Precision	0.98	0.98	1	0.80	0.96	0.75	0.92	0.92	0.86	0.75	0.79	0.99	0.94	0.97	0.96	0.96	0.95	94.57%

Fig. 3 Predicting domain of networks using only four simple structural features. A model is learned using $D = 4$ simple and computationally efficient features, including density, average degree, assortativity, and maximum k-core. Using only these simple features, the model correctly predicts the domain of 94.57% of the networks. Hence, the model remains highly accurate for predicting the domain of an arbitrary network. These results indicate that networks from different domains are structurally distinguishable even using very simple structural properties. Notably, there exists key structural differences among the network even at the most basic fundamental level

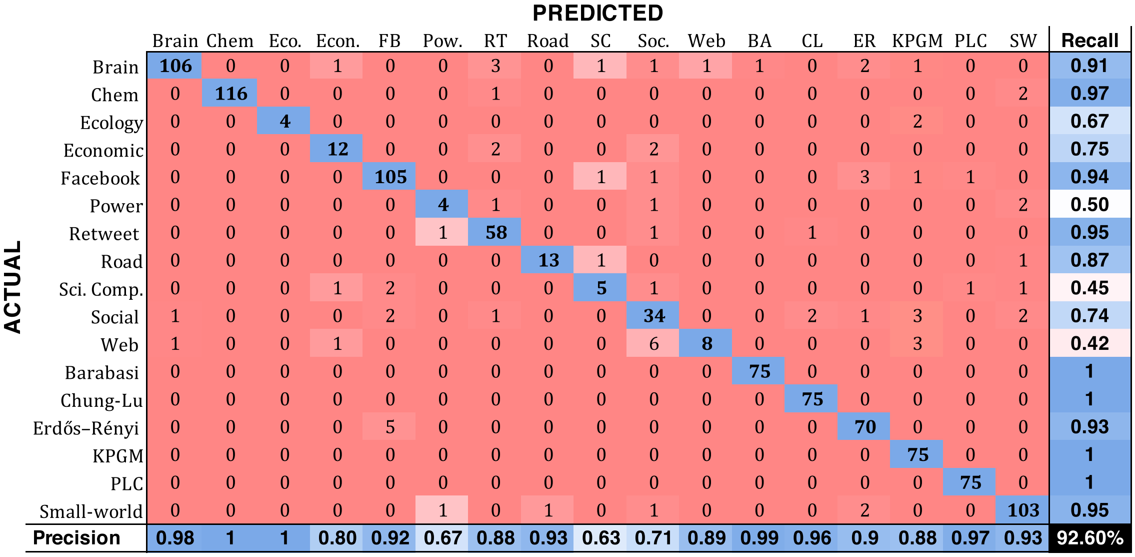
remaining real-world networks from the 11 domains (brain, cheminformatics, ecology, etc.) are relabeled as "real-world networks." This gives us two groups of graphs: real and synthetic. Hence, this task is simply to predict whether a previously unseen graph is a real-world network or not. Using only four simple structural features, the model is able to accurately classify 98.42% of the networks. However, if we use $D = 11$ features, we are able to predict 99.01% of the networks correctly as being either synthetically generated or a real-world complex network from one of the 11 network domains. Interestingly, only three synthetic graphs were incorrectly classified as real, whereas 13 real graphs were incorrectly classified as synthetic. The above is from the model that uses $D = 4$ features. This result illustrates the extent that these synthetic graph models fail to generate realistically looking graphs. For instance, even using four simple structural features, we are able to accurately distinguish whether a graph is synthetic or not.

The goal of many synthetic graph generators is to generate graphs that are indistinguishable from actual real-world networks (Mahadevan et al. 2007; Wang et al. 2007; Moreno et al. 2010; Leskovec et al. 2010; Rossi et al. 2013). As such, many synthetic graph generators are designed such that the synthetic graphs (and their structural properties such as the distribution of degrees and triangle counts) closely resemble graphs from a specific category/domain such as social networks (Leskovec et al. 2010; Rossi et al. 2013). Unfortunately, this experiment demonstrates that not only do these graph models fail to generate graphs that appear realistic, but they are all easily detected as synthetic using extremely simple structural features. These results also hint at a better and more robust approach for evaluating synthetic graph generators. Previous work has mainly focused on generating graphs that preserve specific graph properties or distributions (e.g., degree, triangle). However, as a first step, it would be better to first evaluate whether the graphs generated are indistinguishable from real networks using the simple structural features above. Since if the synthetic graphs are easily classified as synthetic graphs (as opposed to real-world networks) using such simple structural features, then certainly whether the model preserves a specific property or distribution that is substantially more complex is less relevant.

Now, we investigate whether the category/domain of networks can be predicted using only three or four simple graph features.

Result 3 *Networks from different domains/categories are structurally distinguishable using only a few basic structural features* (Figs. 3 *and* 4). *A few simple features are sufficient to accurately predict the category of networks.*

Figure 3 shows prediction results for a model learned with only four simple and computationally efficient graph features. Notably, we learn a random forest model using only four features to characterize each graph, including density $\rho(G)$, average degree d_{avg}, assortativity $r(G)$, and maximum k-core number $K(G)$ as features ($\mathbf{F}_1$–$\mathbf{F}_4$). Despite using only these four features to describe each network, the predictive



PREDICTED

ACTUAL

	Brain	Chem	Eco.	Econ.	FB	Pow.	RT	Road	SC	Soc.	Web	BA	CL	ER	KPGM	PLC	SW	**Recall**
Brain	**106**	0	0	1	0	0	3	0	1	1	1	1	0	2	1	0	0	**0.91**
Chem	0	**116**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	**0.97**
Ecology	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0	2	0	0	**0.67**
Economic	0	0	0	**12**	0	0	2	0	0	2	0	0	0	0	0	0	0	**0.75**
Facebook	0	0	0	0	**105**	0	0	0	1	1	0	0	0	3	1	1	0	**0.94**
Power	0	0	0	0	0	**4**	1	0	0	1	0	0	0	0	0	0	2	**0.50**
Retweet	0	0	0	0	0	1	**58**	0	0	1	0	0	1	0	0	0	0	**0.95**
Road	0	0	0	0	0	0	0	**13**	1	0	0	0	0	0	0	0	1	**0.87**
Sci. Comp.	0	0	0	1	2	0	0	0	**5**	1	0	0	0	0	0	1	1	**0.45**
Social	1	0	0	0	2	0	1	0	0	**34**	0	0	2	1	3	0	2	**0.74**
Web	1	0	0	1	0	0	0	0	0	6	**8**	0	0	0	3	0	0	**0.42**
Barabasi	0	0	0	0	0	0	0	0	0	0	0	**75**	0	0	0	0	0	**1**
Chung-Lu	0	0	0	0	0	0	0	0	0	0	0	0	**75**	0	0	0	0	**1**
Erdős–Rényi	0	0	0	0	5	0	0	0	0	0	0	0	0	**70**	0	0	0	**0.93**
KPGM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**75**	0	0	**1**
PLC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**75**	0	**1**
Small-world	0	0	0	0	0	1	0	1	0	1	0	0	0	2	0	0	**103**	**0.95**
Precision	**0.98**	**1**	**1**	**0.80**	**0.92**	**0.67**	**0.88**	**0.93**	**0.63**	**0.71**	**0.89**	**0.99**	**0.96**	**0.9**	**0.88**	**0.97**	**0.93**	**92.60%**

Fig. 4 Prediction results using $D = 3$ simple structural features including average degree, assortativity, and maximum k-core ($\mathbf{F}_1$–$\mathbf{F}_3$). The model accurately predicts the domain of most networks achieving an accuracy of 92.6%. These results indicate that networks from different domains/categories are structurally distinguishable using a few important fundamental structural features

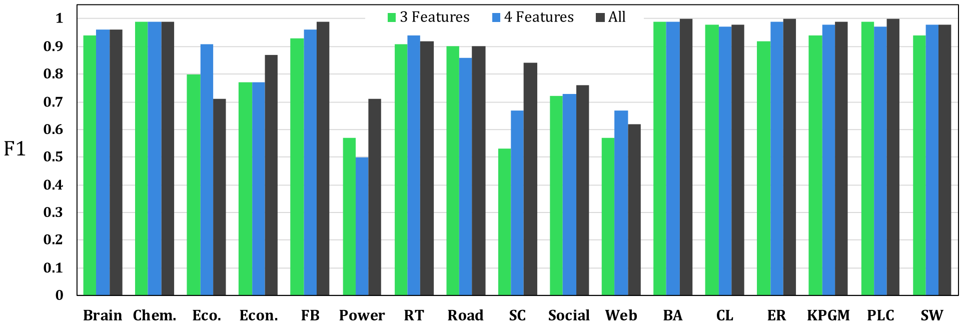


Fig. 5 F1 score results for different complex network domains

model is able to accurately predict with 94.57% accuracy the category (domain) of arbitrary networks. Hence, the model remains highly accurate for predicting the domain of arbitrary networks. Furthermore, the difference in classification performance compared to the previous model using 11 features is small. These results indicate that networks from different domains are structurally distinguishable even using very simple structural properties. Notably, there exists key structural differences among the networks even when characterized using only four simple structural properties.

In addition, Fig. 4 shows prediction results from a model learned using only $D = 3$ features to describe each network. In particular, the features used to characterize each network include average degree, assortativity, and maximum k-core ($\mathbf{F}_1$–$\mathbf{F}_3$). The model achieves an accuracy of 92.6%. Hence, the model remains highly accurate and able to predict the domain of most networks. These results indicate that networks from different domains/categories are structurally distinguishable using only three structural features. We also investigated models with even fewer features. However, in all cases, the overall accuracy decreases significantly, e.g., from 92.6% to around 84%.

Importantly, the results in Figs. 3 and 4 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 $\mathcal{O}(|E|)$. Notice the difference in accuracy, recall, and precision compared to Fig. 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 six different synthetic graph models.

F1 score results for each network category are shown in Fig. 5. From these results, there are three main findings. First, synthetic graphs from any generator are easy to

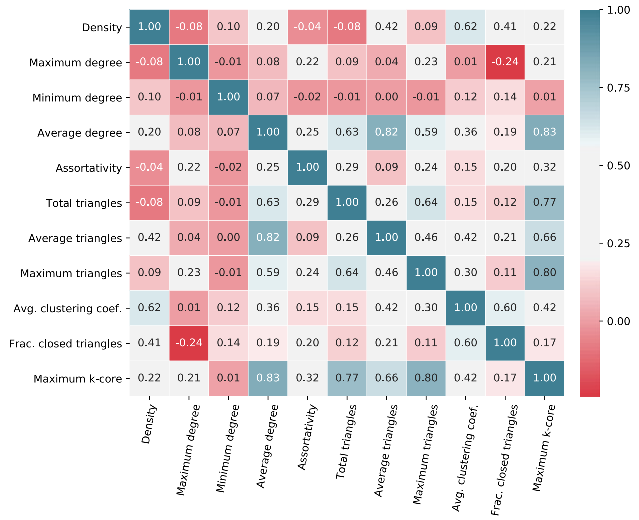


Fig. 6 Structural feature correlations. We measure pairwise Pearson correlation between each pair of structural features $\mathbf{C} = \Phi\langle \mathbf{x}_i, \mathbf{x}_j \rangle,\ \forall i,j$, where $\mathbf{C}$ is a $D \times D$ symmetric correlation matrix and Φ is the Pearson correlation function

classify as the graphs generated from any of the generators are significantly different structurally from any other synthetic graph generated by another model as well as any other real-world network category/domain. Second, most complex networks are easy to correctly predict their domain/category even when using only four simple structural features of the graphs. Third, the simpler model sometimes outperforms the other that uses all available features (e.g., web graphs).

To gain further understanding of the previous classification results, we analyze the possible correlations between the features as follows. 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 and Φ is a similarity function which in this case is Pearson correlation. The correlation matrix is shown in Fig. 6, where 1 is a positive linear correlation, 0 is no correlation, and -1 is negative correlation. In Fig. 6, we observe 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 average degree d_{avg} and the maximum k-core number of a graph G denoted by $K(G)$. Future work will explore whether better classification results can be achieved when replacing average degree with a less correlated feature that is still simple and computationally efficient.

5 Conclusion

This work 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 95.7% accuracy using a simple random forest model to predict the domain and/or generative process governing the formation of the network (Fig. 1). More strikingly, a model learned using only four simple structural features is able to accurately predict the domain of 94.5% of the networks (Fig. 3). 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.

We also find that synthetic graphs are trivial to classify as the model can predict with near-certainty whether a graph is synthetic or not but more importantly the network model used to generate it. This result requires careful consideration as it implies that using synthetic graphs for evaluation, as done previously, should only be carried out with extreme caution. Moreover, this finding also highlights the limitations inherent in common graph models and graph generation algorithms. Since synthetic graph models are generally intended to replicate features in real networks, the observations made in our work highlight the difficulty these models have in creating graphs that appear similar to any of the categories of real-world networks we investigated.

Future Work and Applications The results and findings of this work have a variety of practical applications beyond expanding the understanding of real-world complex networks and synthetic graph models. One important application is to predict the best method (e.g., lowest error, fastest) to use for a specific problem based on the underlying structural properties of the graph. Indeed, the performance of graph algorithms depends largely on the structural properties of the underlying graph of interest. Recent research has focused on identifying the methods that perform best for specific network categories, e.g., designing approximation algorithms that perform best for social networks. For instance, some work has focused on grouping methods for crawling/network sampling methods (Ahmed et al. 2014), maximum and k-clique problem (Rossi et al. 2014), coloring (Rossi and Ahmed 2014), among others. Given the mapping of methods to specific network categories (that consist of graphs with similar structural properties), we can then use the results and findings of our work to predict the category of the network and therefore predicting the method that is likely to perform best for graphs with similar underlying structural properties. In other words, given a previously unseen network and a problem of interest, we can predict the category of the network using the previous model f and then select the method that performs best for graphs with similar underlying structural properties.

In addition, 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 (Rossi and Ahmed 2015a). 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 (Rossi and Ahmed 2015a), 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 computationally efficient structural properties from G denoted by $\mathbf{y}$ and then derive $\mathbf{r} = \mathcal{K}(\mathbf{y}, \mathbf{x}_i)$, for $i = 1, 2, \ldots, |\mathcal{G}|$, where $\mathcal{G}$ is the set of graphs in the graph database (e.g., all graphs available at NR (Rossi and Ahmed 2015a)); $\mathcal{K}$ 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 order the graphs from most similar to least by sorting $\mathbf{r}$ and output the top-k graphs that closely resemble the input graph G provided by the user.

One important direction for future work is to further explore and understand why graphs from synthetic generators are trivial to classify. These results and findings will be important for building better synthetic generators. Future work should also investigate other synthetic graph generators and complex networks from other domains/categories. Furthermore, another open question is whether there is a smaller set of features that can achieve comparable or better predictive performance. Finally, future work should also explore other features and classification models.

Acknowledgements We thank all the reviewers for many helpful suggestions and feedback. We also want to thank Jean-Louis Lassez, James P. Canning, Emma E. Ingram, Sammantha Nowak-Wolff, Adriana M. Ortiz, Karl R. B. Schmitt, Sucheta Soundarajan, and Aaron Clauset for useful discussions and feedback.

References

Abrahao B, Soundarajan S, Hopcroft J, Kleinberg R (2012) On the separability of structural classes of communities. In: KDD

Ahmed NK, Neville J, Kompella R (2014) Network sampling: from static to streaming graphs. TKDD 8(2):7:1–7:56

Ahmed NK, Neville J, Rossi RA, Duffield N (2015) Efficient graphlet counting for large networks. In: ICDM

Ahmed NK, Neville J, Rossi RA, Duffield N, Willke TL (2016) Graphlet decomposition: framework, algorithms, and applications. Knowl Inf Syst (KAIS) 50(3):1–32

Ahmed NK, Rossi RA (2015) Interactive visual graph analytics on the web. In: ICWSM, pp 566–569

Ahmed NK, Rossi RA, Zhou R, Lee JB, Kong X, Willke TL, Eldardiry H (2017) Representation learning in large attributed graphs. In: WiML NIPS

Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74:47

Ali W, Wegner AE, Gaunt RE, Deane CM, Reinert G (2016) Comparison of large networks with sub-sampling strategies. Sci Rep 6:28955

Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin

Bonner S, Brennan J, Kureshi I, Theodoropoulos G, McGough A (2016) Efficient comparison of massive graphs through the use of graph fingerprints. In: KDD MLG workshop

Bonner S, Brennan J, Theodoropoulos G, Kureshi I, McGough AS (2016) Deep topology classification: a new approach for massive graph classification. In: IEEE BigData, pp 3290–3297

Bonner S, Brennan J, Theodoropoulos G, Kureshi I, McGough AS (2016) GFP-X: A parallel approach to massive graph comparison using spark. In: IEEE Big Data, pp 3298–3307

Breiman L (2001) Random forests. Mach. Learn. 45(1):5–32

Canning JP, Ingram EE, Nowak-Wolff S, Ortiz AM, Ahmed NK, Rossi RA, Schmitt KRB, Soundarajan S (2017) Network classification and categorization. arXiv:1709.04481

Canning JP, Ingram EE, Nowak-Wolff S, Ortiz AM, Ahmed NK, Rossi RA, Schmitt KRB, Soundarajan S (2018) Predicting graph categories from structural properties. arXiv:1805.02682

Chung F, Lu L (2002) Connected components in random graphs with given expected degree sequences. Ann Comb 6(2):125–145

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP (2015) Convolutional networks on graphs for learning molecular fingerprints. In: NIPS, pp 2224–2232

Erdős P, Rényi A (1960) On the evolution of random graphs. Publ Math Inst Hungar Acad Sci 5:17–61

Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. Springer, New York

Gärtner T (2003) A survey of kernels for structured data. SIGKDD Explor 5(1):49–58

Gärtner T, Flach P, Wrobel S (2003) On graph kernels: Hardness results and efficient alternatives. learning theory and kernel machines. pp 129–143

Goldsmith TE, Davenport DM (1990) Assessing structural similarity of graphs

Graph500. http://graph500.org/?page_id=12

Guo T, Zhu X (2013) Understanding the roles of sub-graph features for graph classification: an empirical study perspective. In: CIKM. ACM, pp 817–822

Holme P, Kim BJ (2002) Growing scale-free networks with tunable clustering. Physl Rev E 65(2):026107

Ikehara K (2016) The structure of complex networks across domains. PhD thesis, University of Colorado at Boulder

Ikehara K, Clauset A (2017) Characterizing the structural diversity of complex networks across domains. arXiv preprint arXiv:1710.11304

Khan AM, Gleich DF, Pothen A, Halappanavar M (2012) A multithreaded algorithm for network alignment via approximate matching. In: HPCC, pp 64

Kollias G, Sathe M, Schenk O, Grama A (2014) Fast parallel algorithms for graph similarity and matching. J Parall Distrib Comput 74(5):2400–2410

Koutra D, Tong H, Lubensky D (2013) Big-align: fast bipartite graph alignment. In: ICDM, pp 389–398

Kriege N, Mutzel P (2012) Subgraph matching kernels for attributed graphs. arXiv preprint arXiv:1206.6483

Lee JB, Rossi R, Kong X (2017) Deep graph attention model. In arXiv:1709.06075

Leskovec J, Chakrabarti D, Kleinberg J, Faloutsos C, Ghahramani Z (2010) Kronecker graphs: an approach to modeling networks. JMLR 11(Feb):985–1042

Li G, Semerci M, Yener B, Zaki MJ (2012) Effective graph classification based on topological and label attributes. Stat Anal Data Min 5(4):265–283

Mahadevan P, Hubble C, Krioukov D, Huffaker B, Vahdat A (2007) Orbis: rescaling degree correlations to generate annotated internet topologies. ACM SIGCOMM Comput Commun Rev 37(4):325–336

Mahé P, Ueda N, Akutsu T, Perret J-L, Vert J-P (2004) Extensions of marginalized graph kernels. In ICML, pp 70

Malod-Dognin N, Pržulj N (2015) L-graal: Lagrangian graphlet-based network aligner. Bioinformatics 31(13):2182–2189

Milenković T, Ng WL, Hayes W, Pržulj N (2010) Optimal network alignment with graphlet degree vectors. Cancer Info. 9:121

Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827

Moreno S, Kirshner S, Neville J, Vishwanathan S (2010) Tied Kronecker product graph models to capture variance in network populations. In: Proceedings of the 48th annual Allerton conference on communication, control, and computing. pp 1137–1144

Newman M (2010) Networks: an introduction. Oxford Unv. Press, Oxford

Newman ME (2002) Assortative mixing in networks. Phys Rev Lett 89(20):208701

Newman ME, Strogatz SH, Watts DJ (2001) Random graphs with arbitrary degree distributions and their applications. Phys Rev E 64(2):026118

Onnela J-P, Fenn DJ, Reid S, Porter MA, Mucha PJ, Fricker MD, Jones NS (2012) Taxonomies of networks from community structure. Phys Rev E 86(3):036104

Ralaivola L, Swamidass SJ, Saigo H, Baldi P (2005) Graph kernels for chemical informatics. Neural Netw 18(8):1093–1110

Raymond JW, Gardiner EJ, Willett P (2002) Rascal: calculation of graph similarity using maximum common edge subgraphs. Comput J 45(6):631–644

Rossi RA, Ahmed NK (2014) Coloring large complex networks. Soc Netw Anal Min 4(1):1–37

Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: AAAI http://networkrepository.com

Rossi RA, Ahmed NK (2015) Role discovery in networks. TKDE 27(4):1112–1131

Rossi RA, Fahmy S, Talukder N (2013) A multi-level approach for evaluating internet topology generators. In: IFIP networking, pp 1–9

Rossi RA, Gleich DF, Gebremedhin AH, Patwary MA (2014) Fast maximum clique algorithms for large graphs. In: WWW

Rossi RA, Zhou R, Ahmed NK (2017) Deep feature learning for graphs. In arXiv:1704.08829, pp 1–11

Shervashidze N, Schweitzer P, Leeuwen EJV, Mehlhorn K, Borgwardt KM (2011) Weisfeiler-lehman graph kernels. JMLR 12:2539–2561

Shervashidze N, Vishwanathan S, Petri T, Mehlhorn K, Borgwardt K (2009) Efficient graphlet kernels for large graph comparison. In: AISTATS, pp 488–495

Soundarajan S, Eliassi-Rad T, Gallagher B (2014) A guide to selecting a network similarity method. In: SDM, pp 1037–1045

Ugander J, Backstrom L, Kleinberg J (2013) Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In: WWW, pp 1307–1318

van Steen M (2010) Graph theory and complex networks. 1st ed

Vishwanathan S, Schraudolph N, Kondor R, Borgwardt K (2010) Graph kernels. JMLR 11:1201–1242

Wang X, Liu X, Loguinov D (2007) Modeling the evolution of degree correlation in scale-free topology generators. In: INFOCOM

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

Zager LA, Verghese GC (2008) Graph similarity scoring and matching. Appl Math Lett 21(1):86–94

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.