Classification of Synthetic Networks

Sibo Song, Selin Damla Ahipasaoglu, Ngai-Man Cheung, Justin Ruths

Abstract—We explore a new paradigm for establishing similarities between networks using a data-driven approach. Synthetic networks admit ground-truth class labels that enable us to systematically study the supervised classification of synthetic networks, using features determined by established network statistics. In the face of wide variation in the parameters that characterize the network models as well as the contamination of the network structure, simulated through mixed network models of synthesis, our experiments reveal that the support vector machine is able to correctly classify over 98% of the networks. We analyze the experiment results by performing feature selection to gain insights into the discriminativeness and robustness of individual network statistics. We connect the outcome of this study with the existing knowledge about these network models and their expected statistical properties. We complement this analysis with results and analysis from unsupervised approaches, which reveal that classifying network without using any label information is significantly more difficult. We make available our network datasets and program code to facilitate future work.

Index Terms—Synthetic networks, network statistics, supervised classification, feature selection, unsupervised classification.

1 Introduction

NETWORK science has largely followed a scientific approach, namely focusing on developing new statistics with which to characterize networks and new (in-silico) experiments to generate observations. In this work, we leverage this large corpus of work to turn the study of network science around and study networks from a data-driven perspective. In the past two decades, studying systems as networks has provided tremendous insight into various domains such as the organization and robustness of some of our most critical infrastructures, including the properties of the Internet, [1], [2]; the competition and financial stability of global markets through the ties between corporations [3]; the role of transcription factors in coordinating protein regulatory networks [4]; extracting functional clusters within the brain [5]; the rich interactions present in both online and offline social interactions, such as political affiliation, dynamics of the blogosphere, and the spread of happiness [6], [7], [8]; and extinction in ecosystems [9].

Significant interest lies in understanding the causal network structures that identify and differentiate networks. Establishing that two seemingly different types of networks have an underlying similarity can help us study and understand one network by investigating the other. Importantly, networks that are seemingly very different may exhibit such fundamental similarities. Interest in determining these connections between network types has motivated several significant contributions to network science [10], [11], [12].

These studies were insight driven, meaning that they first require a testable hypothesis and then use experiments to gain evidence towards or against this hypothesis. We assert that a data-driven approach will complement such existing approaches by allowing researchers to establish connections that go beyond our current ability or understanding to predict and formulate hypotheses.

In our data-driven approach, we calculate a select set of network statistics on a collection of networks. These statistics - or features - then characterize the networks in a high-dimensional feature space. We use machine learning methods to partition this feature space into categories with similar features and use this partitioning to classify networks. Although a similar idea was independently explored recently, the research challenges and nuances of this work were largely overlooked and the full suite of potential machine learning methods was not used [13], [14]. Because of this, the opportunity for significant insights was lost. Therefore, we aim to provide a systematic study of this novel approach towards network science.

This work marks the beginning of this systematic study, by way of focusing on classifying synthetic network models, for which we have known models of formation and known network structures that these models generate with high probability. Figure 1 shows the framework of classifying and analyzing synthetic network in the paper. In this work we consider six representative network models: Barabasi-Albert (BA) scale-free network model [1], the Erdos-Renyi (ER) random graph model [15], the biologically derived duplication-divergence (DD) model [16], the local attachment (LA) model [17], the small-world (SW) model [18], and the regular graph (REG) model. Using synthetic networks, we can independently vary certain network attributes (e.g., average degree) while maintaining others (e.g., network size). This level of control is important so that we can create and study the classification problem without interference from unintended network differences. As a simple example, accounting for networks of different size (different number of nodes) requires an element of normalization which is not fully understood. Synthetic networks also have known average properties that can help to validate the distinguishing features we find using our approach. Finally, by mixing different synthetic network models, we study the effect of "contamination" and the robustness of classification to perturbation. Throughout the results on classification, we find that the distinguishing features selected by the algorithms are consistent with the features expected by our knowledge of network science.

2 Related work

In this section, we discuss the prior work related to this paper, covering the subgraph or motif analysis to approach

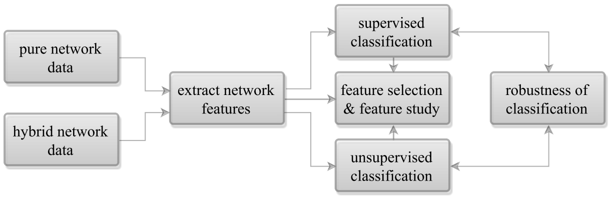

pure network data
hybrid network data
extract network features
supervised classification
feature selection & feature study
unsupervised classification
robustness of classification

Fig. 1. The framework of classifying and analyzing synthetic networks.

network similarity [10], [11], [12], spectral methods for embedding and clustering networks, [19], [20], designing various graph kernels [21], [22], [23], as well as using topological features to classify networks [13], [14].

Motif analysis computes the statistics of the occurrence of small subgraphs, and uses this to measure network similarity. In [10], the authors presented an approach to distinguish the statistically significant (compared to an analogous randomized network) small network structures, or motifs, that are present in various types of networks (e.g., genomic, financial, ecological); in [11], the authors introduced a framework to study the distribution of graphlets, or small subnetworks of a fixed size, as a signature, or distinguishing statistic, of a network; and in [12], the authors uncovered the major causal structures that give rise to the need for a specific number of controls to be added to a network to render it controllable.

Early work in [19], [20] adopts a spectral representation of graphs which generates fixed-length spectral features. Leading eigenvectors of the adjacency matrix are used to compute vectors of spectral properties. Spectral features, therefore, measure the pairwise similarity of the graphs.

Recently, graph kernels are popular for measuring structural similarity of networks. And they map graph features to points in high dimensional inner product spaces, making them amenable to classification techniques such as SVMs. Shortest-path kernel [21] simply compares the sorted endpoints and the length of shortest-paths that are common between two graphs. Weisfeiler-Lehman Subtree kernel [23] decomposes a graph into subtrees and graphlet kernel [22] decomposes a graph into graphlets.

Studies in [13], [14] are most related to our work. They made attempts to tackle the network classification problem using the topological properties. It has been recognized that topologically similar networks are likely to represent systems with functional similarities, whereas network classes with specific topological properties probably have unique functional features. However, the studies in [13], [14] were not comprehensive and thorough, e.g., supervised learning techniques were not used, the real-world networks were selected from relatively small datasets. Also, there was a lack of in-depth analysis to study the relations between topological measures and interpret the classification results. We address these issues in our systematic and comprehensive experiments and in-depth analysis.

3 Approach

Ultimately, we aim to develop a methodology that allows us to classify real-world networks into functionally meaningful categories based on their network structures. This paper describes a first major step and a proof of concept towards this goal by studying the classification of undirected, unweighted synthetic networks. While investigating synthetic networks is certainly easier than real-world networks, we show there is a significant amount of nuance even in this case. Synthetic networks are easier because we can control the types of variation present in the networks we generate. Most importantly, synthetic networks come with ground-truth model types against which we can compare. As we have discussed, synthetic networks can be grouped based on the generative models under which they were created. On the contrary, existing categories of real networks do not exist. Therefore, it is difficult to validate and analyze the results using real networks. Nevertheless, the results on synthetic networks provide some important insights for real networks. Therefore, in this work, we distinguish between networks using a set of network statistics in order to predict the generative model of a sample network. The sample network can be formed using one of six generative models (BA, ER, DD, LA, SW, REG) or some combination of them.

3.1 Network Features and Statistics

The set of features (see Table 1) that we use to train and classify the networks is a collection of representative statistics used in network science to study networks [24]. For a network with a vertex set V, where $n = |V|$, we define the statistics as follows.

Betweenness Centrality [25]. Betweenness centrality is the normalized fraction of shortest paths going through a node u,

$$BC(u) = \sum_{v,w \in V} \frac{\sigma_{vw}^{u}}{\sigma_{vw}}, \tag{1}$$

where σ_{vw} is the total number of shortest paths between nodes v and w, and σ_{vw}^{u} the number of shortest paths between v and w going through u.

Clustering Coefficient [26]. The local clustering coefficient is a statistic on each node, expressing the ratio of pairs of neighbors that are connected to the number of pairs of neighbors. This fraction quantifies the prevalence of triangles in the network.

Closeness Centrality. The closeness centrality of a node measures the average distance from a node to the other nodes in the network. In particular, it is the inverse of the average distance between a node of interest (u) and all the other nodes $v \in V$,

$$CC(u) = \frac{n-1}{\sum_{v \in V} d_{uv}}. \tag{2}$$

Degree. The degree of a node in a network is the number of edges that connect to the node. The degree distribution is a frequency histogram of the number of nodes that have a particular degree.

Singular Values. These are singular values of the adjacency matrix A, where $A_{ji} = 1$ (we consider unweighted networks) when there is an edge from node i to node j.

The network statistics described above are properties that form a distribution for any given network, with a sample for each node (n samples in total). Some related literature on graph sampling chooses to directly compare these types of distributions between two networks [27].

mean value of Betweenness Centrality	μ_{bc}
standard deviation of Betweenness Centrality	σ_{bc}
minimum value of Betweenness Centrality	m_{bc}
maximum value of Betweenness Centrality	M_{bc}
mean value of Clustering Coefficient	μ_{cl}
standard deviation of Clustering Coefficient	σ_{cl}
minimum value of Clustering Coefficient	m_{cl}
maximum value of Clustering Coefficient	M_{cl}
mean value of Closeness Centrality	μ_{cc}
standard deviation of Closeness Centrality	σ_{cc}
minimum value of Closeness Centrality	m_{cc}
maximum value of Closeness Centrality	M_{cc}
mean value of Degree Distribution	μ_{dd}
standard deviation of Degree Distribution	σ_{dd}
minimum value of Degree Distribution	m_{dd}
maximum value of Degree Distribution	M_{dd}
mean value of Singular Values	μ_{sv}
standard deviation of Singular Values	σ_{sv}
minimum value of Singular Values	m_{sv}
maximum value of Singular Values	M_{sv}
diameter	d

TABLE 1
This study uses a feature set of twenty-one of the most representative network statistics, normalized to zero mean and unit variance.

While this may work well enough in this study, our aim is to build up an approach that generalizes to real world networks. Comparing distributions across networks of different sizes is not fully understood. We would also like to be able to classify networks by calculating statistics only on a subset of the nodes in the network and recovering the entire distribution of a statistic from a sample subset is still a topic of current research. Furthermore, comparing distributions yields a number that represents the error between the two distributions, but does not clarify whether it is because, for example, right-skewed instead of left-skewed or has narrow support versus wide support. Therefore, in this study, we choose to calculate the mean, standard deviation, minimum value and maximum value of these distributions, in order to avoid the ambiguity of comparing distributions directly and prepare this work for extension to real-world networks, where we cannot control network sizes. We add to this list of network statistics one global statistic, diameter.

Diameter. Diameter is the maximum eccentricity value of the network.

$$Diam = \underset{u \in V}{max} \quad e(u), \tag{3}$$

where $e(u)$ denotes the eccentricity or the longest shortest-path between node u and any other node in the network. Therefore, diameter is the longest shortest-path in the whole network.

This list of network features provides twenty-one statistics characterizing each network. We then rescale each of these statistics to have zero mean and unit variance so that none of the features dominates the classification algorithm. This gives us a 21-dimensional feature vector that describes each of the networks we generate.

3.2 Supervised Classification Algorithm

We utilize a Support Vector Machine (SVM) for supervised classification using the 21-dimensional network features. An SVM constructs a hyperplane or set of hyperplanes in the high-dimensional feature space. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called the functional margin), since in general the larger the margin, the lower the generalization error of the classifier [28].

SVM classification is essentially a binary classification technique, which has to be modified to handle the multiclass tasks. A one-against-one approach is used, which involves constructing a classifier for each pair of classes resulting in $N(N-1)/2$ classifiers, where N is the total number of network categories. When applied to a test point, each classification gives one vote to the winning class and the point is labeled with the class having the maximum number of votes. In our experiments, we choose the regularization parameter $C = 10$ for all cases. We apply a linear kernel since it is efficient and performs well in general.

Training and testing. We follow the standard practice to train and test the models. If the same dataset supplies both the training data and test data, we use 10-fold cross validation and report the average classification accuracy. Otherwise, one dataset is used exclusively for training and the other dataset exclusively for testing.

3.3 Unsupervised Classification Algorithm

For unsupervised classification (or clustering), we train the classifier model without using any ground-truth network type labels. We use four popular unsupervised approaches: k-means [29], spectral clustering [30], Gaussian Mixture Model (GMM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [31].

The k-means approach is arguably the most popular unsupervised classification approach which is also a special case of a GMM. It clusters data by trying to separate samples in n groups of equal variance, minimizing sum-of-squares of all pairs of distances within each cluster. Spectral clustering makes use of eigenvalues and eigenvectors of the Laplacian matrix of the data points. A GMM approach implements the expectation-maximization (EM) algorithm for fitting mixtures of Gaussian models to predict the labels of data. The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. It utilizes radius and the minimum number of points required to form a dense region to group the data points. It does not require one to specify the number of clusters and also it can find arbitrarily shaped clusters which makes it very flexible.

Note that the evaluation of the performance of an unsupervised classification algorithm is different from a supervised classification algorithm. In this work, we choose the V-measure [32] as the evaluation metric, which can be understood using the concepts of homogeneity and completeness. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. We assume a data set comprising N data points, and two partitions of these: a set of classes, $C = \{c_i \mid i = 1, 2 \ldots, n\}$ and a set of clusters, $K = \{k_i \mid i = 1, 2 \ldots, m\}$. Let A be the contingency table produced by the clustering algorithm representing the clustering solution, such that $A = \{a_{ij}\}$ where a_{ij} is the number

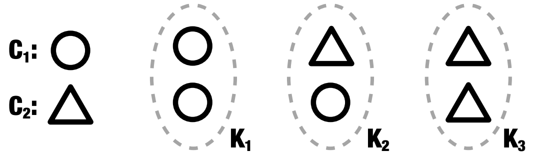


Fig. 2. An example of an unsupervised classification result.

of data points that are members of class c_i and elements of cluster k_j. Homogeneity is defined as

$$h = \begin{cases} 1 & \text{if} \quad H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else,} \end{cases} \tag{4}$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|} \frac{a_{ck}}{N} \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}, \tag{5}$$

$$H(C) = -\sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{ck}}{n} \log \frac{\sum_{k=1}^{|K|} a_{ck}}{n}, \tag{6}$$

and completeness is defined as

$$c = \begin{cases} 1 & \text{if} \quad H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else.} \end{cases} \tag{7}$$

$H(K|C)$ and $H(K)$ are defined in a symmetric manner. The V-measure is defined as the harmonic mean of homogeneity and completeness

$$v = \frac{2hc}{h+c}. \tag{8}$$

For example, as illustrated in Fig. 2, six data points in two classes are classified into three clusters. $H(C|K) = -(\frac{2}{6}\log\frac{2}{2} + \frac{2}{6}\log\frac{2}{2}+\frac{1}{6}\log\frac{1}{2} + \frac{1}{6}\log\frac{1}{2}) = -\frac{1}{3}\log\frac{1}{2}$, and $H(C) = -(\frac{2}{2}\log\frac{2}{2} + \frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}) = -\log\frac{1}{2}$, then we will have homogeneity $h = 1 - \frac{H(C|K)}{H(C)} = 0.67$. Similarly, completeness is $c = 0.42$. V-measure is therefore given as $v = \frac{2hc}{h+c} = 0.51$.

3.4 Feature Selection and Feature Study

We perform feature selection to identify the most discerning network features. In practice, with a large number of features, we have the risk of over-fitting and this could lead to a significant drop in classification accuracy. Thus, it is advantageous to reduce the number of features used. Optimal feature selection requires examining every feature combination and is intractable with even a moderate number of features. Many sub-optimal approaches have been proposed in the literature. Here we use two greedy approaches, forward feature selection and mRMR (minimal-redundancy-maximal-relevance criterion) [33] for feature selection. Forward feature selection iteratively adds features that achieve the maximum increase in the accuracy for a given classifier. mRMR uses mutual information to model the relevancy and redundancy of the feature and uses the model to choose features. In addition, we study the relevance between features and networks and examine the correlation between network features.

4 Network Datasets

In this section, we describe the design of our synthetic network datasets for classification experiments. We first discuss the design of a *pure* network dataset in which each network is generated by a single network model. Then, we discuss a *hybrid* network dataset in which each network is generated by multiple network models.

4.1 Pure Networks

We select six different network models to be discussed in detail. This selection includes models that are the most represented in the literature. These models employ formation mechanisms that provide a diverse set of characteristics: sequential and non-sequential, random and deterministic, preferentially and uniformly selected, application inspired and theoretical. There are also intentional overlaps in formation mechanisms between these models so that we can observe how this would affect the classification performance.

In this work we fix the size of the network to be $n = 1000$ nodes and generate 200 networks of each type such that each model is equally represented. For each type we vary the parameters that characterize the network model such that the average node degree (k) of these 200 networks increases linearly from (on average) two edges per node ($k = 2$) to twenty edges per node ($k = 20$). This represents an extremely wide range of average degree, so we consider this dataset to encompass a much larger variation than is typically observed in real networks.

- *Barabasi-Albert (BA)* [1]. The BA model is characterized by preferential attachment and a scale-free degree distribution. The model initially seeds the network with m nodes and then sequentially introduces one new node at a time, connecting edges between the new node and the existing nodes in a preferential fashion biased proportion to node degree. The value of m can be calculated from the desired average degree such that $m = (N - \sqrt{N^2 - 4kN})/2$.
- *Duplication-Divergence (DD)* [16]. The DD model is characterized by a vertex copying method as well as a scale-free distribution for some parameter values. The model begins with two connected nodes. At each step an existing node is chosen and copied, meaning that the new node has edges to the neighbors of the copied node. These edges are kept with a probability s (at least one edge must remain, otherwise the node is discarded and the process is repeated). There are some results on the relationship between the parameter s and the average degree, however, we employed a more accurate method by computing this relationship empirically.
- *Erdos-Renyi (ER)* [15]. The ER model is the classic random network characterized by uniform random connection and a Poisson degree distribution. The model randomly connects any two nodes with a probability p.
- *Local Attachment (LA)* [17]. The LA model is characterized by a scale-free degree distribution that shares many similar properties with BA networks, however, introduces an element of clustering. The model begins with m fully connected nodes and at each step

pure model	
BA	200
DD	200
ER	200
LA	200
REG	200
SW	200
total	1200

hybrid contamination probability	1%	5%	10%	20%	30%
BA	60	60	60	60	60
DD	60	60	60	60	60
ER	60	60	60	60	60
LA	60	60	60	60	60
				total	1200

TABLE 2
Number of synthetic networks in our pure and hybrid network datasets.

	BA	DD	ER	LA	REG	SW
BA	100					
DD		100				
ER			99.24			0.76
LA		0.05		99.95		
REG			0.39		99.56	0.05
SW			10.19			89.81

TABLE 3
The confusion matrix with average classification accuracy of 98.17%. Row labels indicate the ground truth and entries identify the percentage of this type of network that was classified as each of the column labels.

introduces a new node with m new connections to the existing nodes. Of these m edges, m_r are attached randomly (uniformly) to existing nodes. The remaining $m - m_r$ are attached randomly (uniformly) to the neighbors of the m_r nodes previously selected. The same connection between m and average degree exists as in the BA model. Traditionally the LA model is constructed as a directed network; here we simply drop direction on the edges of the synthesized graphs to yield an undirected version. Because this model has two parameters that characterize its formation, the fraction of m_r/m was varied along $[0, 1]$ in eight steps, and the average degree was varied by 25 steps - thereby maintaining 200 total networks of this type.

- *Regular (REG)*. Regular graphs are very rigidly defined networks characterized by a regular lattice formation such that each node has exactly the same degree. Such a simplified model is relevant here because it provides a contrast to other more heterogeneous network types.
- *Small World (SW)* [26]. The SW model, in its most simplified form, is characterized by both local and long-range connectivity. This model is seeded by assembling the nodes in a ring and connecting each node to its q closest neighbors (q must be even). Subsequently, an ER type probabilistic approach - connecting any two nodes with probability p - creates shortcuts across the circle of nodes.

4.2 Hybrid Networks

Synthetic network models are highly stylized versions of reality. True complex networks are not a simple reproduction of a single pure synthetic network model. In order to study the robustness of our classification results with respect to perturbation of the network structure, we construct contaminated, or "hybrid", networks that are built by incorporating multiple network models into one generative method.

There is no standard approach to generate hybrid networks by mixing two or more generative methods. Therefore, here we describe our proposed approach. The hybrid dataset considers BA, DD, ER, and LA type models because they are all generative (sequential) models, or can be extended to sequential models in a straightforward manner, unlike REG and SW network models.

Our approach is able to create hybrids of any two, three, or all network types by placing four probabilities that govern the likeliness that a particular network model is selected. Because we require a ground truth, any mixture should still retain a dominant network model type which has a higher probability than the others. The network is seeded according to the chosen dominant network model. In BA- and LA-dominant networks, m nodes are initially seeded according to their respective methods; in DD-dominant networks two nodes connected to each other are initially seeded; in ER-dominant networks no seed is added. Because these seed sizes are unequal, we then adjust the probabilities for each network type so that the overall sample should be consistent with the original probabilities.

At each step i, for $i = \{1, 2, m, \ldots, N\}$ (recall the seeding procedure may add 1, 2, or m nodes initially), we randomly select the model of formation that should be used by the i^{th} node based on the probabilities assigned to each type: preferential attachment (BA), vertex copying (DD), random connection (ER), or local attachment (LA). This process is well defined for BA, DD, and LA. The ER model is not a sequential model, therefore, we allow the node to potentially make connections with any of the other $N - 1$ nodes, however, for the other models, we only provide access to nodes 1 up to $i - 1$. In this work, we consider mixtures of two network models at a level of 1%, 5%, 10%, 20%, and 30%. A total of 1200 networks are generated, 300 of each dominant network type, 240 of each contamination level. The pure and hybrid network datasets are summarized in Table 2.

5 Supervised Classification

Our goal is to comprehensively understand the ability to classify pure networks into categories that correspond to the generative models that formed them. To begin, we focus on the pure network dataset that aggregates 1200 pure networks, representative of the six different generative models (BA, ER, DD, LA, SW, REG). We train and test a linear SVM on this dataset using 10-fold cross validation. The classification result is shown in Table 3, where the row label is the ground truth (i.e., the generative model that created the network) and the percentages in each row indicate what fraction of the networks generated by individual models were classified as belonging to the network model of the corresponding column label. The average accuracy is 98.17% indicating a very good performance overall. Small-World networks, in particular, are relatively more difficult to classify (bottom row) because the SW model uses a mixed approach where the network is first seeded in a regular-type fashion and then random, ER-like, connections are added. This intuition is reflected in the fact that SW networks are often confused with ER networks.

	BA	DD	ER	LA
BA	100			
DD		100		
ER			100	
LA	0.05	0.14		99.81

(a) Average: 99.95%

	BA	DD	ER	LA
BA	85	5	6.67	3.33
DD	13	75.33	4	7.67
ER	0.33		99.67	
LA	62	3.67	8.33	26

(b) Average: 71.5%

	BA	DD	ER	LA
BA	100			
DD		100		
ER			100	
LA				100

(c) Average: 100%

	BA	DD	ER	LA
BA	97.3	0.27		2.43
DD	1.55	98.45		
ER			100	
LA	1.52	0.68	0.23	97.58

(d) Average: 98.33%

TABLE 4

Confusion tables, in percentages, for (a) testing SVM_{pure} on the pure networks, (b) testing SVM_{pure} on the hybrid networks, (c) testing SVM_{hyb} on the pure networks, (d) testing SVM_{hyb} on the hybrid networks. As discussed, in (a) and (d), since the same dataset supplies training and testing samples, 10-fold cross validation is used and 90% of the dataset is selected for training in each fold, and the averages are reported. In (b) and (c), an entire dataset is used for training and another dataset is used for testing.

It is expected that the classifier would perform well on these pure networks because the models tend to create networks with relatively definable characteristics. However, it should be underscored that we selected very wide ranges of parameter values used in each model. For example, the average degree of these networks ranged from 2 connections per node (on average) to 20 connections per node. Such parameter swings create substantially different networks, however, the algorithm was still able to correctly classify these networks with average accuracy 98.17%.

5.1 Robustness of Classification

Synthetic network models are caricatures of the characteristics we observe in real networks. Therefore, by definition, they do not represent the range of heterogeneity and variety that is present in real-world networks. Therefore, we next analyze whether the classifier, trained on the pure network data would perform as well on contaminated networks generated by a hybrid model, in this case employing two different synthetic network models from BA, ER, DD, LA. The REG and SW models of formation occur in a non-sequential manner, which is incompatible with the sequential formation of BA, ER, DD, and LA (ER is not by nature sequential, however, it can be adapted to fit a sequential mechanism). Because we now constrain ourselves to these four network types, we retrained the SVM on the corresponding networks and yield an average accuracy of 100% (see Table 4a); we will refer to this SVM trained on the four synthetic network types as SVM_{pure}. Contaminated hybrid networks were created for each possible pair of network models with contamination set at five levels: 1%, 5%, 10%, 20%, and 30%.

Table 4b shows the classification result using SVM_{pure} to classify the contaminated hybrid networks. The average accuracy drops to 71.5%, which clearly shows that the SVM trained on pure networks is not robust to perturbations in the network structure. To understand this more clearly, in Figure 3 we show the average performance of SVM_{pure} applied separately to hybrid networks with different amounts of contamination or mixing ratios. We note that with only a 0.01 mixing ratio of contamination (i.e., only 10 nodes out of 1000 are added with a different attachment model), the classification accuracy drops to 78.33%.

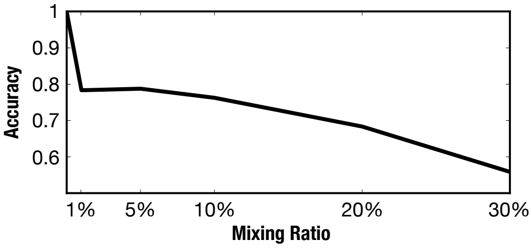


Fig. 3. Accuracy of testing SVM_{pure} on hybrid networks with different levels of contamination.

Figure 4 shows the confusion matrices for the different different mixing ratios. We observe that the performance of the classifier is quite heterogeneous, affecting some network types more than others. In particular, ER networks with any level of contamination are easy to classify whereas DD and LA networks are classified increasingly incorrectly.

5.1.1 *Effect of Different Training Sets*

While it is not a surprise that different training sets yield different classification results, what is compelling is the improvement and the robustness of this performance in the presence of perturbation. In the previous section, we tested the robustness of SVM_{pure} on the hybrid network models and found the classifier to be highly sensitive to even a small amount of contamination. We now train a new SVM, SVM_{hyb}, on the corpus of 1200 hybrid networks using the same procedure as before.

We test SVM_{hyb} on both the pure networks (Table 4c) and hybrid networks (Table 4d). These tables show that the SVM trained on the hybrid networks is able to classify the hybrid networks with excellent accuracy (98.33%), while still classifying the pure networks with near perfect accuracy (100%). Another way to state this is that by training the SVM with a different dataset, we are able to introduce 30% contamination with virtually no loss in classification accuracy. We will spend some time to investigate not only the characteristics of the performance but also the features that facilitate this level of robust classification.

The effect of the training set can be seen more clearly in Fig. 5, for which we have trained five more SVM classifiers: $\text{SVM}_{\text{hyb}[1\%]}$, $\text{SVM}_{\text{hyb}[5\%]}$, $\text{SVM}_{\text{hyb}[10\%]}$, $\text{SVM}_{\text{hyb}[20\%]}$, $\text{SVM}_{\text{hyb}[30\%]}$, trained on only the hybrid networks with 1%, 5%, 10%, 20%, and 30% contamination, respectively. Because these SVMs are trained on datasets with a particular mixing ratio, each will show a peak classification accuracy at that mixing ratio, however, the breadth of that peak indicates the robustness of the classifier. In Fig. 5, the peak broadens as it shifts right, indicating that the SVMs trained on networks with higher contamination are more robust to all levels of contamination.

Rank	$\mathrm{SVM}_{\mathrm{hyb}}$	$\mathrm{SVM}_{\mathrm{pure}}$	$\mathrm{mRMR}_{\mathrm{hyb}}$	$\mathrm{mRMR}_{\mathrm{pure}}$
1	σ_{cl}	σ_{cl}	σ_{cl}	σ_{cl}
2	μ_{cc}	M_{dd}	μ_{cc}	σ_{cc}
3	σ_{cc}	σ_{bc}	M_{dd}	m_{cl}
4	m_{cl}	m_{cl}	m_{dd}	M_{dd}
5	σ_{sv}	μ_{cc}	m_{sv}	μ_{cc}
6	μ_{dd}	μ_{bc}	m_{cc}	M_{cl}
7	M_{bc}	m_{bc}	M_{cc}	m_{cc}

TABLE 5
Comparing the results of the forward feature selection / mRMR for $\mathrm{SVM}_{\mathrm{pure}}$ and $\mathrm{SVM}_{\mathrm{hyb}}$ provide insights into the discriminativeness and robustness (to contamination) of individual features. The features that are in common are connected by links. The features that experience significant change in ranking are highlighted.

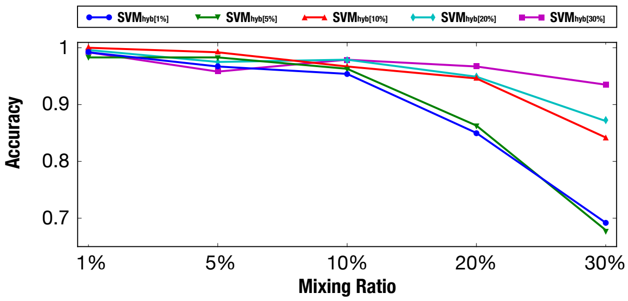


Fig. 5. The contamination level of the networks used to train the classifier distinctively effects its robustness.

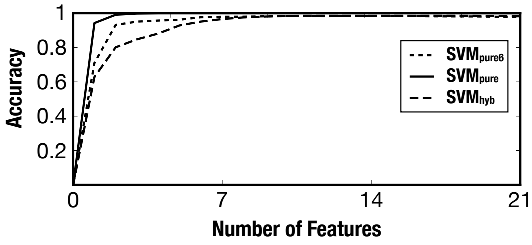


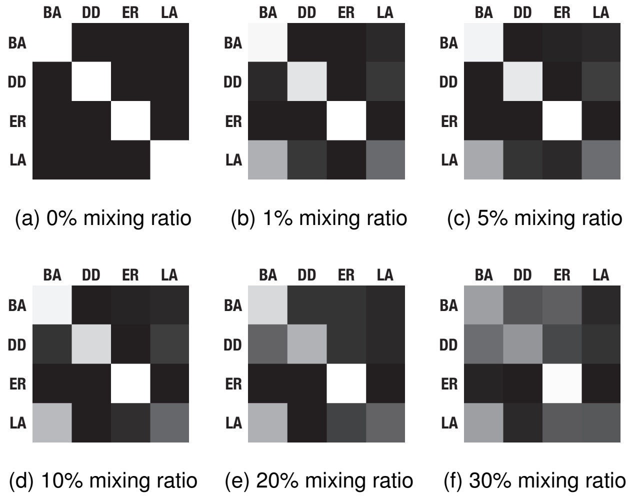
Fig. 6. An increasing number of features (in the order listed in Table 5) are used to improve the accuracy of (a) $\mathrm{SVM}_{\mathrm{pure}}$ and (b) $\mathrm{SVM}_{\mathrm{hyb}}$ classifiers.

5.2 Feature Selection and Analysis

Thus far, we have shown that pure networks can be classified well by an SVM based on standard network statistics. We have shown that this classifier breaks down very quickly when contamination is introduced into the networks; however, by training on the contaminated networks we can recover nearly all of the classification accuracy. While we use the same set of features (network statistics) in both cases, in the hybrid case the classifier establishes relationships between the network categories and the features that are significantly more robust. From the network science perspective, these robust relationships are interesting because we anticipate them to be more generalizable to real-world networks. Here we explore the differences between these classifiers.

We use a forward feature selection mechanism to establish the ranking of important features in both cases. Forward feature selection is the greedy process of choosing one-by-one the next feature which improves the accuracy of classification the most. The process continues, sequentially adding one new feature at a time, in the end producing an ordered list of the features. Note that this approach is non-optimal. For example, if two features are both the most predictive feature (i.e., yielding the highest and equal classification accuracy using each feature by itself) the greedy approach will select only one as the most predictive feature. If in addition the two features are perfectly correlated (i.e., redundant) then the second feature will appear at the very end of the feature ranking, however, this cannot be interpreted as meaning that the feature has very little predictive power. This approach, however, is able to provide very quantifiable relationships between classes and network statistics, some of which, as we show here, echoes our understanding of synthetic network models. Other relationships go beyond our intuitive understanding of these network models and reveal the strength of a machine learning approach to the study of networks. In future work, when these methods are used on real networks, these insights will reveal new relationships for network scientists to investigate.

We extract a ranking of the features for $\mathrm{SVM}_{\mathrm{pure}}$ and $\mathrm{SVM}_{\mathrm{hyb}}$ using this forward feature selection method and the results are summarized in Table 5. Figure 6 shows that the classifier reaches most of its predictive capacity after only the first seven features. In order to help explain some of the feature selection results we use Table 6, which shows the single-feature classification accuracy of each feature to individually classify each network type. For example, the 74.90% in the first row, first column for $\mathrm{SVM}_{\mathrm{pure}}$ in Table 6 indicates that, on average, 74.90% of networks were classified correctly as BA or not-BA using only one feature, μ_{bc}. The two heatmaps in Fig. 7 go one step further and show the overall classification accuracy using two features, where the diagonal entries are the same as the last ("All") column in Table 6. These tables and figure provide an imperfect view of the correlations that exist between the features. We use them to justify some of the observations we make about the



(a) 0% mixing ratio
(b) 1% mixing ratio
(c) 5% mixing ratio
(d) 10% mixing ratio
(e) 20% mixing ratio
(f) 30% mixing ratio

Fig. 4. Confusion matrix for testing $\mathrm{SVM}_{\mathrm{pure}}$ on hybrid networks with different levels of contamination.

	BA	DD	ER	LA	**All**
μ_{bc}	74.90%	74.51%	75.03%	75.43%	29.00%
σ_{bc}	75.64%	75.25%	92.54%	74.29%	52.18%
m_{bc}	85.43%	75.78%	75.38%	74.24%	45.04%
M_{bc}	75.14%	75.90%	94.96%	75.80%	57.45%
μ_{cl}	75.16%	74.36%	75.24%	94.35%	53.08%
σ_{cl}	75.11%	99.75%	75.88%	94.48%	93.88%
m_{cl}	74.59%	75.65%	74.65%	92.13%	57.63%
M_{cl}	74.83%	74.56%	74.71%	94.79%	53.28%
μ_{cc}	75.41%	91.44%	74.58%	97.99%	79.69%
σ_{cc}	75.81%	96.38%	74.96%	98.14%	91.49%
m_{cc}	75.05%	74.43%	75.69%	97.46%	43.51%
M_{cc}	75.36%	96.98%	75.45%	91.65%	70.55%
μ_{dd}	75.35%	75.29%	75.15%	75.53%	25.15%
σ_{dd}	73.80%	84.86%	88.69%	75.15%	57.63%
m_{dd}	75.59%	77.55%	74.44%	75.08%	44.90%
M_{dd}	74.89%	82.40%	92.29%	74.83%	66.61%
μ_{sv}	75.04%	74.36%	74.99%	75.64%	40.26%
σ_{sv}	74.33%	85.01%	75.16%	74.86%	40.01%
m_{sv}	74.98%	75.01%	74.91%	74.21%	38.04%
M_{sv}	75.28%	83.41%	75.89%	75.31%	41.81%
d	74.78%	75.49%	74.65%	75.65%	30.58%

	BA	DD	ER	LA	**All**
μ_{bc}	75.38%	75.03%	75.61%	74.62%	24.47%
σ_{bc}	74.80%	74.57%	88.61%	74.83%	41.51%
m_{bc}	75.25%	75.02%	74.93%	74.33%	35.83%
M_{bc}	75.14%	74.39%	89.76%	75.36%	46.64%
μ_{cl}	74.76%	74.83%	74.74%	75.03%	29.83%
σ_{cl}	74.86%	89.16%	86.79%	74.58%	65.93%
m_{cl}	74.75%	75.16%	75.53%	74.75%	31.17%
M_{cl}	75.48%	74.75%	75.03%	75.59%	28.23%
μ_{cc}	75.31%	75.45%	74.89%	86.78%	48.38%
σ_{cc}	74.63%	75.18%	75.38%	79.49%	37.18%
m_{cc}	74.88%	74.85%	75.84%	77.13%	23.68%
M_{cc}	75.28%	75.11%	75.46%	75.02%	32.56%
μ_{dd}	75.18%	75.17%	75.16%	74.93%	26.31%
σ_{dd}	75.55%	78.44%	77.35%	74.22%	46.24%
m_{dd}	75.24%	75.20%	74.79%	74.94%	38.68%
M_{dd}	75.19%	82.03%	85.73%	75.66%	54.40%
μ_{sv}	75.01%	75.08%	75.58%	75.13%	35.46%
σ_{sv}	74.68%	74.88%	74.60%	74.97%	31.57%
m_{sv}	75.13%	73.27%	74.91%	74.87%	41.88%
M_{sv}	74.50%	75.18%	75.05%	74.64%	34.59%
d	75.36%	74.80%	75.33%	74.10%	27.96%

TABLE 6

Accuracy of individual features on each network type (BA, DD, ER, LA) and overall (All), SVM$_{\text{pure}}$(left) and SVM$_{\text{hyb}}$(right). The accuracies in these tables can be used to highlight the characteristic properties of the network models that become evident in the classification process, such as the fact that the spread of clustering (σ_{cl}) in the pure networks is distinctive for DD (which as no clustering) and LA (which tends to have high clustering) networks.

feature selection, however, because their information is not comprehensive, we do so with care.

The first feature that is selected is common across all classifiers and is the standard deviation of clustering, σ_{cl}. Among the models considered here, clustering is a clearly discriminating network statistic. The LA network formation model is explicitly designed to add clustering to a scale-free model of network formation, because the nominal BA model has low clustering. The vertex copying strategy of the DD model inherently has no clustering because the newly added node is never connected to the node that is being copied, thereby never producing any triangles. The distinction of LA and DD networks can be seen clearly in Table 6 (SVM$_{\text{pure}}$), in which σ_{cl} has high accuracy for LA and DD. In contaminated networks, this feature remains informative for two reasons. At a high level, the standard deviation (and mean) is a more robust statistic than the minimum or maximum values of a statistic (e.g., clustering) because the change in the value of a single node does not dramatically alter the standard deviation of a distribution, but might dramatically change the maximum value of the distribution. More specific to the statistic itself, clustering is a local property in the sense that clustering is determined by looking only at the connectivity with a node's immediate neighbors. Therefore, changes in one node's model of attachment will only effect a few number of nodes around it whereas other non-local statistics (e.g., centralities) depend on a larger set of nodes.

For SVM$_{\text{pure}}$, the maximum of the degree distribution is the second feature, however, it is not robust and its rank drops significantly in the feature selection of SVM$_{\text{hyb}}$. It is not surprising that the feature is important in the pure network dataset. ER networks are likely to have a much smaller maximum degree than other models, because their degree distributions lack the long tails characteristic of scale-free distributions. This is reflected in Table 6 SVM$_{\text{pure}}$, but not to the same level in SVM$_{\text{hyb}}$. This feature lacks robustness because the maximum of a distribution can be more volatile than mean or standard deviation. For example, a single node contamination in an ER dominated network can lead to a very highly connected node, which could easily shift the usually low maximum value of the degree distribution. We also see evidence that contamination reduces the presence of the long tail in the scale-free distributions.

Notice for any feature other than the first selected, that the next feature must not only be discriminating, but also complementary to the other features already selected (although identifying this complementarity is often not immediately obvious). We observe this effect here as M_{dd} provides lower overall discrimination when compared with σ_{cc} (see Table 6, SVM$_{\text{pure}}$), yet is selected as the second feature, seemingly due to the fact that M_{dd} provides separating power for ER networks, whereas σ_{cc} separates DD and LA networks - something that the first statistic, σ_{cl} already does.

For SVM$_{\text{hyb}}$, the second feature selected is μ_{cc}, the mean value of the distribution of closeness centrality. This feature also appears as the fifth feature for SVM$_{\text{pure}}$. The bright spot in the SVM$_{\text{hyb}}$ heatmap in Fig. 7 clearly indicates the high accuracy of using μ_{cc} in combination with σ_{cl}. While it is possible to motivate the mean of closeness centrality as being a robust feature for hybrid network classification from the differences between the different network models, its interaction with σ_{cl} is not obvious. We can observe the fact that constructing a classifier manually would be extremely challenging by looking at Fig. 8. We can see that the patterns in the histograms of each network type over the entire range of σ_{cl} and μ_{cc} exhibit fine-grain differences. For example BA, ER, and LA exhibit the coarse level pattern that most networks of these types have relatively low σ_{cl} and high μ_{cc}. Within this coarse pattern, the hybrid classifier defines more fine-grain relationships that segment the network models into different classes. We emphasize the fact that the broad

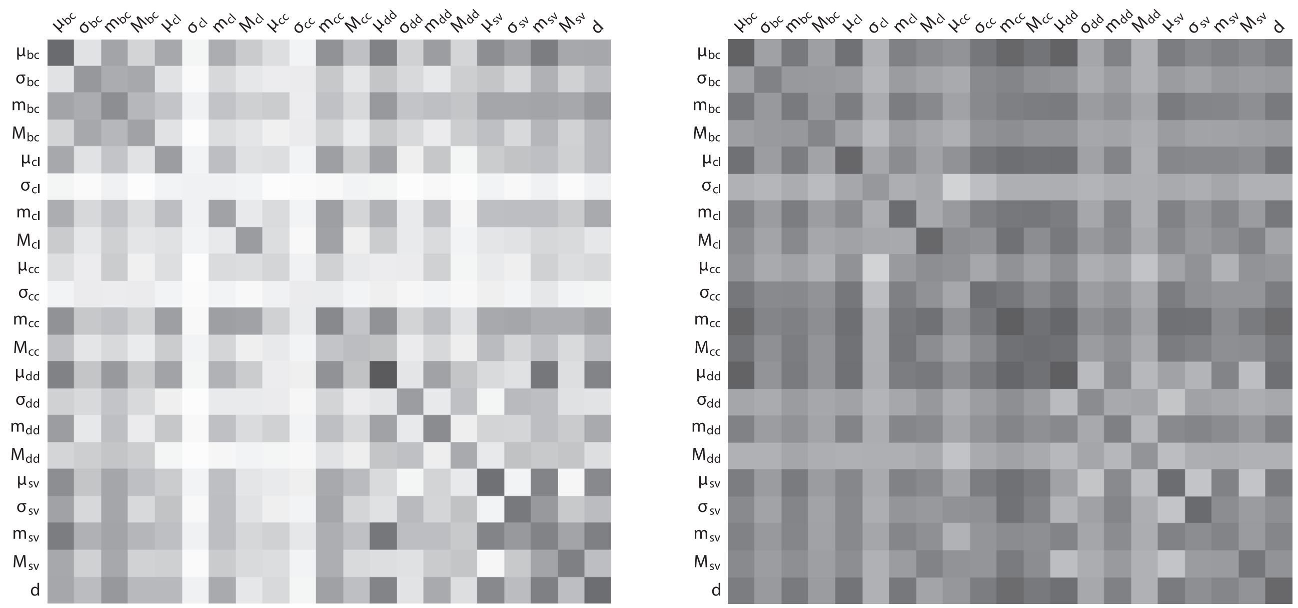
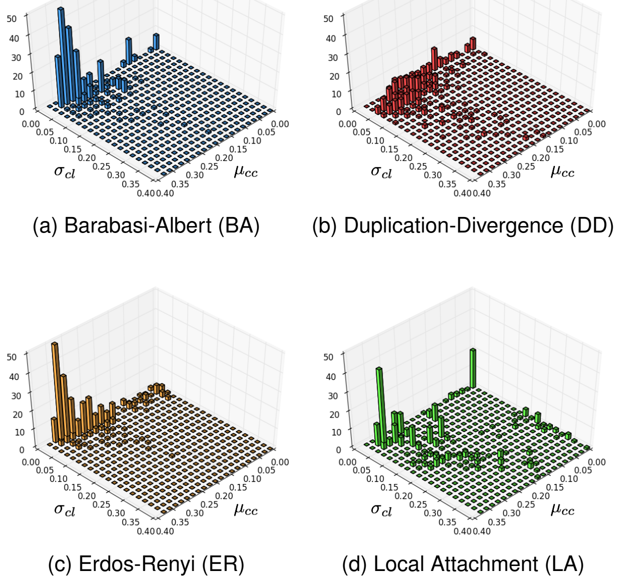


Fig. 7. Feature correlation heatmaps indicate the accuracy of classification using one feature (diagonal elements) and using two features (the heatmaps are inherently symmetric) for SVM$_{pure}$(left) and SVM$_{hyb}$(right). These provide an imperfect picture of the correlation between features. White and black represent 100% and 0% accuracy, respectively.

sampling of networks across very different densities and the wide range of contamination leads to the broad diffuse nature of the patterns in Fig. 8.

The feature selection results observed here tend to agree with the definitions and consequences of the different network models we consider. The SVM classifiers clearly go beyond our ability to categorize networks because the relationships are based on more subtle differences that arise from the data. We do foresee that feature selection can, however, guide network scientists to dig deeper into certain relationships to discover substantive relationships between network models or, in the future, different types of real-world networks. Researchers can make use feature selection to inform which plots, such as Fig. 8, they should construct and evaluate.



(a) Barabasi-Albert (BA)

(b) Duplication-Divergence (DD)

(c) Erdos-Renyi (ER)

(d) Local Attachment (LA)

Fig. 8. Histograms of hybrid network data provide the pattern that represents each network type according to the first two selected features, σ_{cl} and μ_{cc}. The relationships that separate the network models are based on relatively fine-grain details of these patterns.

5.2.1 Max-Relevance and Min-Redundancy Approach

Max-Relevance and Min-Redundancy selects features based on mutual information. The goal is to select a feature subset set that best characterizes the statistical property of a target classification variable, subject to the constraint that these features are mutually as dissimilar to each other as possible, but marginally as similar to the classification variable as possible [33].

By applying mRMR on our pure network dataset, we obtain results in Table 5. We can observe that mRMR also chooses σ_{cl} as the most important feature, and this is consistent with forward feature selection. By comparing the feature lists from the two approaches, we observe some discrepancies. We believe this is due to the fact that forward feature selection gives the second correlated feature a low ranking. For example, σ_{cc} is the second feature from mRMR result and this individual has an overall accuracy of more than 90%. However, σ_{cc} and σ_{cl} are also correlated by examining Table 6: they have very similar performances for all network categories, which is not the case of M_{dd}.

It is not surprising that different methods for ranking features yield different results; it is well-known that there may exist many quasi-equivalent subsets of features that yield similar levels of performance. Since most feature selection methods use certain heuristics, they are not very powerful in distinguishing between the minor differences in classification performance. Future work can experiment recent approaches developed to enumerate all quasi-equally informative subsets [34].

6 Unsupervised Classification

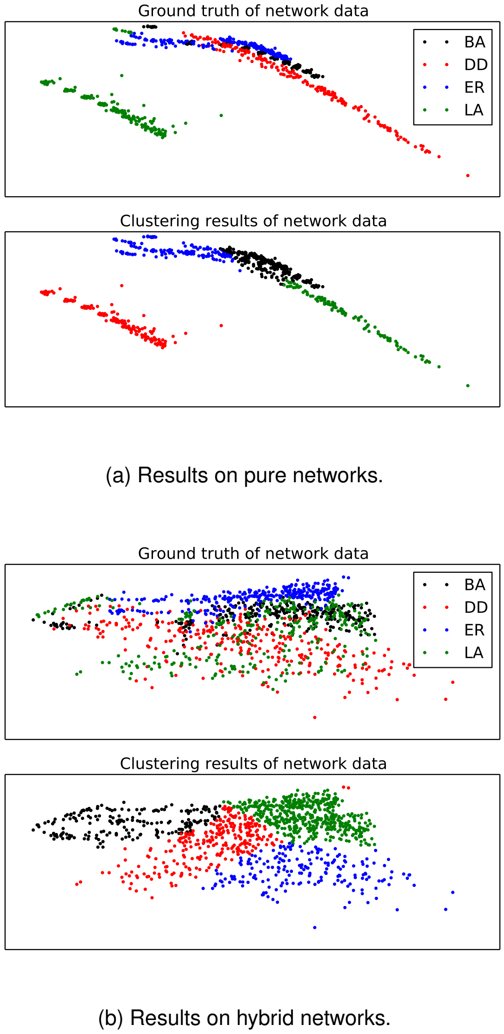
In many applications, real-world networks come with no definite model labels. Therefore, we also investigate unsupervised classification, where we learn the classifier models and cluster the data without using any model label information. Essentially, unsupervised classification learns the hidden structure from the unlabeled data to determine the closeness or distributions of data points for clustering the data. Unsupervised classification is certainly a more difficult task.

To evaluate the unsupervised approaches on both synthetic and hybrid networks, we choose the synthetic and hybrid networks of BA, DD, ER and LA. As before, networks are represented as 21-dimensional feature vectors of network statistics. Note that individual network statistics are correlated, therefore, the intrinsic dimensionality of the feature vectors tends to be much smaller than their dimensionality. As discussed in Section 3.3, we use V-measure as the metrics of unsupervised performance.

The result is provided in Table 7. It shows that spectral clustering, GMM and DBSCAN perform better than k-means on almost all the cases with carefully selected parameters. Density-based approaches like spectral clustering and DBSCAN could achieve better V-measure but they require careful parameter tuning. The performance of k-means is inferior. We believe the reason is that k-means makes the assumption that clusters are convex and isotropic, which is not the case for networks. GMM is more flexible and can approximate the data distribution with Gaussian models of arbitrary shape. Spectral clustering utilizes an affinity matrix using the nearest neighbors method. Similarly, DBSCAN is a density-based clustering algorithm which clusters data according to different density of data points. Table 7 reveals that unsupervised classification of networks is a rather challenging problem and is very sensitive to the purity of the data. A small amount of contamination (1% or 5%) could significantly degrade the classification performance.

To provide some insight of the clustering results, we visualize the projection of the feature vectors onto a 2-dimensional sub-space. Figure 9 displays the data distribution and also the clustering results of pure and hybrid networks. We reduce the dimensionality to two using PCA and then classify these data using the GMM approach. Note that the colors of labels in the clustering results do not necessarily correspond to the (hidden) ground truth, since the GMM clusters data regardless of the value of labels. It confirms that the data are correlated in the space. It shows that hybrid networks data is hardly clustered using these two features. In particular, with contamination, we can observe that the statistics could be mixed up in high dimensional space. This explains the poor performance on clustering hybrid networks in Table 7.

It is tempting to conclude that the success of the supervised classification results indicates that classification of synthetic networks is a relatively trivial task. We present the unsupervised case here to emphasize that the classification task for synthetic networks is not a simple one, despite the extremely high performance achieved through supervised learning. The full treatment of unsupervised network classification requires significantly more study, and is a fruitful direction for future work.



(a) Results on pure networks.

(b) Results on hybrid networks.

Fig. 9. An illustration for unsupervised classification in two dimensional subspace.

7 Conclusion

This paper described a systematic and comprehensive study of the classification of synthetic networks. We generated a network dataset based on six widely-used network models. We identified twenty-one network statistics and used a supervised learning approach with SVM. We examined the effect of contamination in classifiability of networks and investigated features that are discriminative and robust to contamination. Our key findings include:

- Pure synthetic networks are readily classifiable even though they are generated using a wide range of parameters in our experiments.
- Hybrid networks are classifiable if the classifier is carefully trained with network samples that are sufficiently contaminated. Such trained classifier appears to be fairly robust and works well across a range of contamination rates.

	homogeneity	completeness	V-measure
k-means	0.570	0.590	0.580
spectral clustering	0.714	0.772	0.742
GMM	0.714	0.744	0.729
DBSCAN	0.981	0.677	0.801

(a) On pure networks (BA, DD, ER, LA)

	1%	5%	10%	20%	30%
k-means	0.239	0.224	0.198	0.164	0.097
spectral clustering	0.638	0.378	0.443	0.191	0.137
GMM	0.352	0.278	0.227	0.224	0.076
DBSCAN	0.513	0.477	0.417	0.272	0.146

(b) On hybrid networks (V-measure)

TABLE 7
Performance of different unsupervised approaches on pure and hybrid networks.

- Forward feature selection results reveal that the classifiers reach full capability with only a few network features. Thus, it is advantageous to perform some feature selection to reduce the dimension of the feature vectors.
- Standard derivation of clustering coefficient is identified as the most discriminating feature for both pure synthetic networks and hybrid networks. This feature and other relevant features agree with intuition gained from the network science literature on the characteristic statistics of these network models.
- The difficulty of unsupervised classification highlights that network classification is a challenging task, despite the high classification accuracy in the supervised case.

While the aim of unsupervised classification of real networks is still several steps away, our results here on synthetic networks help to clarify the challenges in achieving this goal. Synthetic network models provide intuitive stereotypes to study in order to hone methods from machine learning for the purposes of establishing similarity between networks through network classification.

References

[1] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[2] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[3] S. Vitali, J. B. Glattfelder, and S. Battiston, "The Network of Global Corporate Control," *PLoS ONE*, vol. 6, no. 10, p. e25995, 2011.

[4] H. Yu and M. Gerstein, "Genomic analysis of the hierarchical structure of regulatory networks," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 14 724–14 731, 2006.

[5] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, pp. 186–198, 2009.

[6] M. A. Porter, P. J. Mucha, M. E. J. Newman, and C. M. Warmbrand, "A network analysis of committees in the U.S. House of Representatives." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 7057–7062, 2005.

[7] M. Gotz, J. Leskovec, M. McGlohon, and C. Faloutsos, "Modeling Blog Dynamics," in *Proceedings of the International Conference on Weblogs and Social Media*, 2009.

[8] J. H. Fowler and N. A. Christakis, "Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study," *BMJ*, vol. 337, p. a2338, 2008.

[9] R. E. Ulanowicz, C. Bondavalli, and M. S. Egnotovich, "Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 97: The Florida Bay Ecosystem," Tech. Rep. UMCES-CBL 98-123, 1998.

[10] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[11] N. Przulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 26, no. 6, pp. 853–854, Mar. 2010.

[12] J. Ruths and D. Ruths, "Control profiles of complex networks." *Science (New York, N.Y.)*, vol. 343, no. 6177, pp. 1373–6, Mar. 2014.

[13] B. Kantarci and V. Labatut, "Classification of complex networks based on topological properties," in *Cloud and Green Computing (CGC), 2013 Third International Conference on*. IEEE, 2013, pp. 297–304.

[14] A. Duma and A. Topirceanu, "A network motif based approach for classifying online social networks," in *Applied Computational Intelligence and Informatics (SACI), 2014 IEEE 9th International Symposium on*. IEEE, 2014, pp. 311–315.

[15] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[16] I. Ispolatov, P. Krapivsky, and A. Yuryev, "Duplication-divergence model of protein interaction network," *Physical Review E*, vol. 71, no. 6, p. 61911, 2005.

[17] M. O. Jackson and B. W. Rogers, "Meeting strangers and friends of friends: How random are social networks?" *The American economic review*, pp. 890–915, 2007.

[18] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, pp. 440–442, 1998.

[19] B. Luo, R. C. Wilson, and E. R. Hancock, "Spectral feature vectors for graph clustering," in *Structural, syntactic, and statistical pattern recognition*. Springer, 2002, pp. 83–93.

[20] ——, "Spectral clustering of graphs," in *Graph Based Representations in Pattern Recognition*. Springer, 2003, pp. 190–201.

[21] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[22] N. Shervashidze, T. Petri, K. Mehlhorn, K. M. Borgwardt, and S. Vishwanathan, "Efficient graphlet kernels for large graph comparison," in *International conference on artificial intelligence and statistics*, 2009, pp. 488–495.

[23] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *The Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[24] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[25] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1979.

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[27] J. Leskovec and C. Faloutsos, "Sampling from large graphs," *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 631–636, 2006.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[30] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[32] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 410–420.

[33] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[34] G. Karakaya, S. Galelli, S. D. Ahipasaoglu, and R. Taormina, "Identifying (quasi) equally informative subsets in feature selec-

tion problems for classification: A max-relevance min-redundancy approach," *Cybernetics, IEEE Transactions on*, 2015.

Sibo Song He received Bachelors degree in Automation from Zhejiang University in 2013. He is currently a Ph.D. student at Information Systems Technology and Design Pillar, Singapore University of Technology and Design with research interests in egocentric video processing, deep learning and graph mining.

Selin Damla Ahipaşaoğlu is an Assistant Professor at the Singapore University of Technology and Design since 2012. She received her Ph.D. in 2009 from Cornell University and held research positions at Princeton University and London School of Economics. Her main research focus is developing algorithms for large scale optimization, in particular first-order methods for convex formulations. She is also working on applications in image processing, statistical learning, optimal experimental design, and discrete choice modeling.

Ngai-Man Cheung is an assistant professor at the Singapore University of Technology and Design. He received his Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles. His research interests include image / signal processing and classification with applications to health-care, network science and cyber-security.

Justin Ruths Justin Ruths is an assistant professor at the Singapore University of Technology and Design. Justin holds degrees in Physics (BS, Rice University), Mechanical Engineering (MS, Columbia University), Electrical Engineering (MS, Washington University in Saint Louis), and Systems Science and Applied Math (PhD, Washington University in Saint Louis). His research themes include casting problems in the natural sciences and medicine as optimal control problems and investigating the control of large-scale systems. Towards this latter goal, some of his recent work is at the interface of control and network science.