On Classifying Complex Networks by their Topological Features

Marina von Steinkirch
Department of Physics & Astronomy
Stony Brook University, NY, USA

ABSTRACT

The study of *complex networks* pervades all of the sciences. Characterizing complex network's structure is a key to understand any unifying principles underlying their topology. Previous works have shown that many topological properties can vary for different types of system. However these works generally focus only on a few characteristics at time. In this work we present methods and results for an extensive analysis of 20 global and local graph topological features of 1245 publicly available networks. The raw networks can have orders ranging from a few hundred nodes (*e.g.*, some small biological and small ego-centered examples) to hundred of thousand nodes (*e.g.*, roads and large ego-centered examples). In order to perform the classification task, we sample them into five sets of different graph orders, assigning each of them to one of following four classes: *technological networks*, *information networks*, *biological networks*, and *social networks*. We then perform a comprehensive classification analysis, using several supervised and unsupervised methods, and one-vs.-all/one-vs.-one approaches. As a result, we are able to report remarkable testing set accuracies larger than 90% for the majority of the approaches and set configurations. Additionally, we confidently identify whose are the key topological features for complex network classification.

1. INTRODUCTION

A complex network can be defined as a set of interacting elements possessing some emerging properties, which only appears when considering the system as a whole. The structure and dynamics of complex networks are intrinsically related, since structure always affects function. The study of such systems are modeled with graphs, in which elements and their relations are represented by nodes and links, respectively.

With the emergence of graph databases, various graph kernel methods have been proposed for the task of classifying sub-graphs. However, these methods have been proven to have high computational overhead due to the combinatorial nature of graphs. When it comes to classifying entire networks, a more suitable approach is to consider that graphs belonging to the same class have similar topological descriptions and label attributes. The main idea is to associate a *feature vector* to each graph so that it enables the access to any learning machine developed for statistical feature vectors.

Graph classification is an important data mining task that aims to learn a discriminative model from training examples and then using the model to predict class labels of testing examples. Graphs can be characterized by many different measures. Authors have focused on several properties of networks which are able to represent a range of different system (for example, see [1], [2], [3], [4], [5], and [6]). In this work, we report the methods and results for complex networks classification based on 20 global and local graph topological features. Our database is composed by five sampled sets for different order numbers: $n = 100, 300, 500, 1000$, and 2000. Each of them containing three samples of up to 1245 network publicly available databases [7] [8] [9] [10].

The rest of this paper is divided as follows. In the section II, we review each of the 20 topological features and the 4 class labels for the complex networks classification. In the section III, we describe the data extraction and sampling. In the section IV, we present the classification results, which are discussed in the section V.

2. THEORETICAL INTRODUCTION

The graph selection problem can be stated as follow: There is a dataset of M graphs $G_i \in \mathcal{D}$, with $i = 1, ..., M$. Each graph $G_i = (V_i, E_i)$ is given as a collection of vertices, $V_i = \{v_{i1}, ..., v_{in}\}$ and edges $E_i = \{(v_a, v_b)|v_a, v_b \in V_i]\}$. The graph G_i may have features on the nodes and/or edges, drawn from some common set of j features Σ for the entire data set $\mathcal{D}$. Finally, each graph G_i has a corresponding class $y_i \in C$, where C is the set of categorical class labels, given as $C = \{1, ..., l\}$ ($l = 4$ here). The goal of graph classification is to learn a model $f : \mathcal{D} \rightarrow C$ that predicts the class label for any graph. The model is learned from a training set of graphs with known class labels and then evaluated on a testing set of graphs. The accuracy of the classification model can be tested by comparing the predicted output $\hat{y}_i = f(G_i)$ with the true class label y_i.

In the following subsections, we first describe the $j = 20$ graph topological attributes used in this work as the classi-

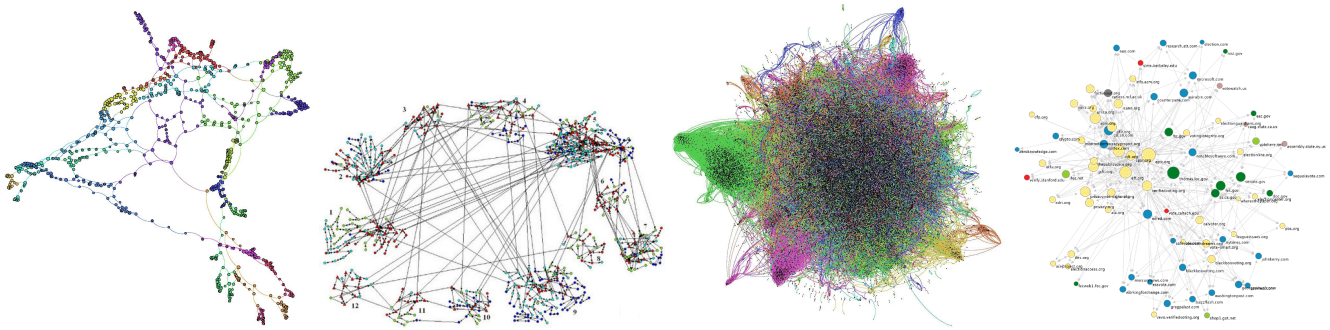

Figure 1: Snapshots of some of the networks in this work: Texas road system (tech), yeast protein interaction (bio), Brightklite location based network (social), and US patent citation (info) [7] [8] [9] [10]

fication features. We then point out the four chosen class labels for complex networks, as suggested in [11]. These sets of feature vectors $F_j = (f_{j1}, ..f_{jM})$ and their corresponding class labels are then used to construct the classifiers in the following sections.

2.1 Topological Features

For each graph $G_i \in \mathcal{D}$, we calculate the following topological features:

1. **Order(Ord) and Size (Siz)**: Respectively, the total number of nodes, $n = |V|$, and the total number of edges, $m = |E|$, in the network.

2. **Betweenness Centrality (Bet)**: For a *node* u, it is the sum of the fraction of all-pairs shortest paths that pass through u. If we denote σ_{vw} as the total number of shortest paths between v and w, and $\sigma_{vw}(u)$, the total number of shortest paths between nodes v and w going though u, the betweenness centrality is $bc(u) = \sum_{v<w\neq w} \frac{\sigma_{vw}(u)}{\sigma_{vw}}$.

3. **Closeness Centrality (Cen)**: For a *node* u, it represents the reciprocal of the average shortest path length between u and every other reachable node in the graph: $cc(u) = \frac{n-1}{\sum_{v\in\{V_u\}} d(u,v)}$, where $d(u,v)$ is the length of the shortest path between the nodes u and v. It measures how fast information spreads from a given node to other reachable nodes in the graphs.

4. **Degree (Deg)**: For a *node* u, it is defined as the number of its neighboring edges. It can be formally defined using the *adjacency matrix*: $deg(u) = \sum_{v\in V} a_{uv}$. In real-world networks, the average degree often follows a power law (*scale-free* networks).

5. **Eccentricity (Ecc)**: For a *node* u, it represents the maximum length of the shortest path between u and every other node in G. If u is isolated, then $ecc(u) = 0$.

6. **Clustering coefficient (Clu)**: For a *node* u, it represents the likelihood that any two neighbors of u are connected: $clc(u) = \frac{2e}{k_u(k_u-1)}$, where k_u is the number of neighbors of u and e_u is the number of connected pairs of neighbors. If all the neighbors nodes of u are connected, then, the neighborhood of u is complete and $clc = 1$. If no nodes in the neighborhood of u are connected, $clc = 0$.

7. **Square clustering coefficient (Scl)**: While *clc* gives the likelihood that any two neighbors of u are connected, *scc* gives the probability that two neighbors of node v share a common neighbor different from v.

8. **Pagerank (Pag)**: It is a ranking of the nodes in the graph G based on the structure of the incoming links.

9. **Communicability centrality (Com)**: For a *node* u, it is the sum of closed walks of all lengths starting and ending at node u.

10. **Coreness (Cor)**: A k-core is a maximal subgraph that contains nodes of degree k or more. The *cor* of a node is the largest value k of a k-core with that node.

11. **Density (Den)**: It is the ratio of existing to possible links in G. It ranges from no link at all to all nodes connected (0 and 1 respectively): $den(G) = \frac{m}{n(n-1)}$. Real networks are usually very sparse, with ~ 0.1.

12. **Maximum Effective Eccentricity or Diameter (Dia)**: It represents the maximum value of *ecc* over all nodes in the graph.

13. **Minimum Effective Eccentricity or Radius (Rad)**: It represents the minimum value of *ecc* over all nodes in the graph G.

14. **Assortativity (Ass)**: It measures the similarity of connections in G with respect to the node degree. Graphs that have only single edges between vertices tend (in the absence of other biases) to show disassortative mixing by degree because the number of edges that can fall between high-degree vertex pairs is limited. Since most networks are represented as simple graphs this implies that most should be disassortative [12].

15. **Number of Cliques (NCl)**: A clique in an undirected graph is a subset of its vertices such that every two vertices in the subset are connected by an edge.

16. **Number of Triangles (NTr)**: The number of triangle connections in G.

17. **Clique number (Cnu)**: The size of the largest clique for G.

18. **Transitivity (Tra)**: A global measure of *clc*, it computes the fraction of all possible triangles present in G. The transitivity ranges from 0.1 to 0.8 in the real world network. It can be interpreted as the probability for two neighbors of a node to be connected.

19. **Edge connectivity (Eco)**: The minimum number of edges that must be removed to disconnect G.

2.2 Types of Networks

We label each of the complex networks into four classes :

- ⋆ **Social Networks** are sets of people or groups of people with some pattern of interactions. An important feature in these networks is the so-called *small-world* experiment [13]. They also often contain a few number of hubs (vertices with high degree). Examples: ego centered networks, population records, affiliations.
- ⋆ **Information (Knowledge) Networks** correspond to data linked together. Examples: citation networks (which are acyclic networks), www (web pages and their links, peer to peer, keyword index.
- ⋆ **Technological Networks** are human-made networks designed for distribution of some commodity or resource, such as electricity or information. Examples: the internet (physical network of computers), radio, telephone, power grids, road systems.
- ⋆ **Biological Networks** are nature-based networks. Their edges are usually symmetrical and directed. Examples: metabolic pathways, generic regulatory network, food web, neural networks.

3. DATA PROCESSING

Data Collecting

We collected 1245 complex networks from four publicly available databases: SNAP [7], KONECT [8], ND [9], and VLADO [10]. These network are labeled as the following:

- ⋆ **Social Networks**: 5 social networks with geographic checkins, 983 ego-centered (Twitter, g+, Facebook), 4 Signed networks with positive and negative edges (friend/foe).
- ⋆ **Information (Knowledge) Networks**: 2 Citation (nodes represent papers, edges represent citations), 8 Collaboration (nodes represent scientists, edges represent collaborations), 3 Communication (email communication networks with edges communication), 4 Webgraphs (nodes represent webpages, edges are hyperlinks), 4 Amazon Product Review, 9 Peer-to-peer.
- ⋆ **Technological Networks**: 117 Autonomous systems (graphs of the internet), 7 Roads (nodes represent intersections and edges roads connecting the intersections).
- ⋆ **Biological Networks**: 2 Carbon exchanges, 43 Cellular (substrate in cellular networks), 43 Metabolic networks (interactions between enzymes and metabolites), 3 Yeast (protein-protein interaction), 8 Atlas (food-webs).

Data Sampling

To be able to use networks of different sizes, we need to sample the large graph to get a smaller similar graph, *i.e.* a smaller graph that preserves the properties of the original graph as much as possible. This is the *scale-down goal*. For this purpose, we sample each graph with five order numbers (number of nodes): 100, 300, 500, 1000, and 2000. The sampling algorithm is based on *snowball sampling*, where a graph starts with some set of seed nodes of interest, and then repeatedly adds some neighbors of the seed nodes and their incident edges. We assume that these networks and their samples follow approximate power law distribution. In general, the snowball sampling tends to underestimate this power law exponent due to bias towards high degrees nodes. It also can underestimates assortativity and node/link sampling [14] [15] [16]. As a consequence, we see in the next sections, we see that features such as assortativity have less influence in the networks with smaller order.

Due to the probabilistic character of the sampling, we choose to extract 3 samples each time. This produces a data set with three times of the total number of all of the graphs that have enough nodes to be include in each of the five order sets. The sampling algorithm is also robust against results containing too many isolated nodes or too few edges. The resulting graph orders are within %10 of the nominal order.

Data Feature Extraction

For each of the five sets of sampled graphs we extract the 20 topological features described in the previous section. The full resulting datasets by either dividing by features or dividing by order are available publicly at [18]. The software for feature extraction task and data cleansing are made available as open- source at [19] and [20].

Data Standardization

Standardization of datasets is a common requirement for many machine learning estimators. If a feature has a variance that is orders of magnitude larger that others, it might dominate the objective function and make the estimator unable to learn from other features correctly [17].

In this work, after separating 20% of the data for testing set, we perform three parallel approaches. The first leaves the data **without** any standardization ('none'). This allows to check the validity of any standardization in the many classifiers, and whether the use of the non-standardized data is sufficient.

In the first standardization method ('gauss') we ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature. We then scale it by dividing non-constant features by their standard deviation.

The second standardization method ('xmin') is given by scaling the features to lie between a given minimum and maximum value, often between zero and one. The motivation to use this scaling includes the robustness to very small standard deviations of features and the preservation of zero entries in sparse data.

In the Figs. 2 and 3 we can see the correlation plots for the sets with order 100, 300, 1000, and 2000, for their best scoring features (as we see in the following section). In general, the standardization preserves the distribution of the data, Its effect on the classifiers is discussed in the following section.

4. CLASSIFICATION

Adaboost

We perform **one vs. all** classification for each of the four class labels using **Adaboost** with **decision stumps**. In this case, for each of the classes, we separately calculate their binary classification, by setting all the other three classes as

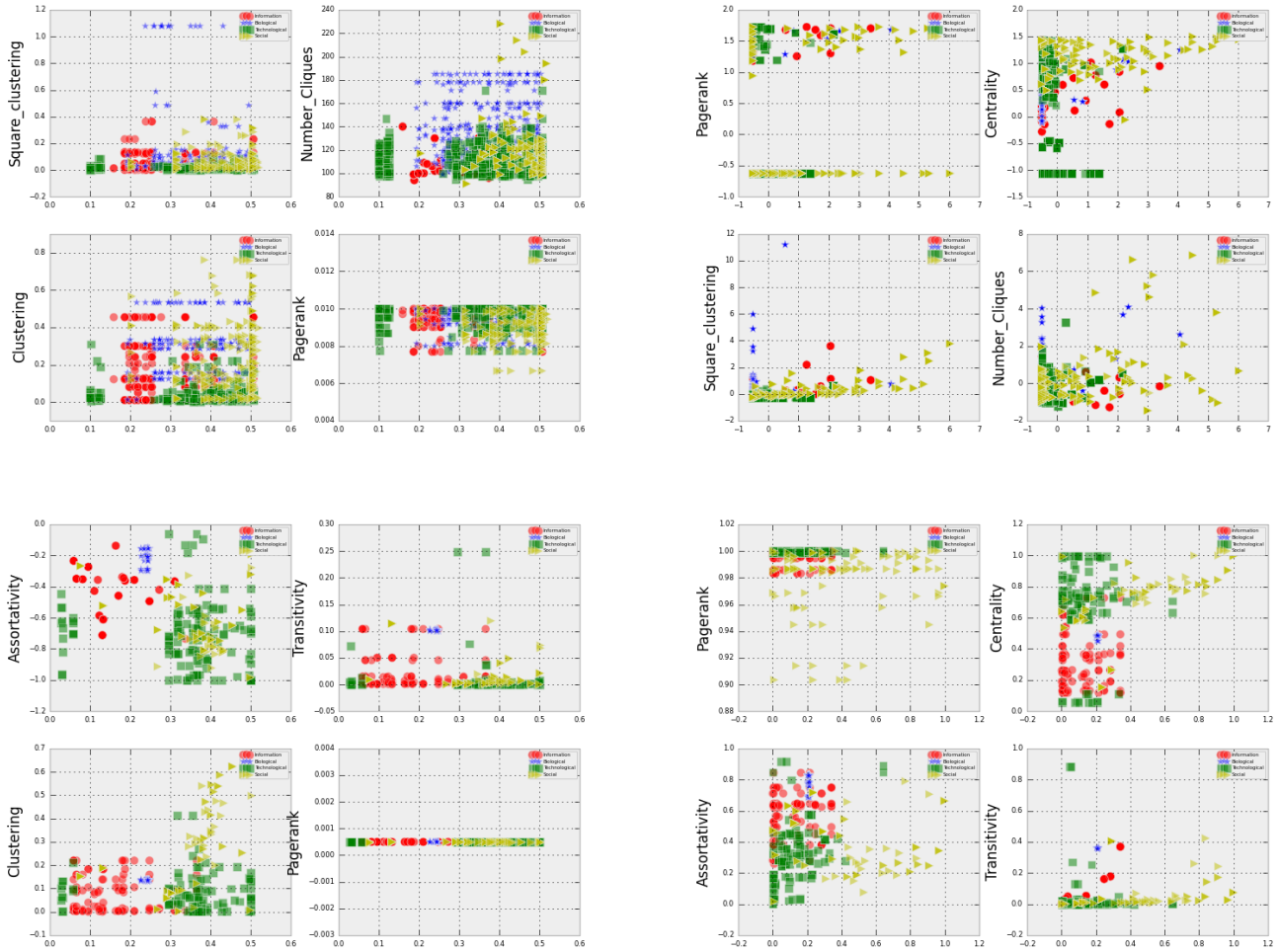

Figure 2: **In the left side we see correlation plots for Centrality, with no standardization, for the for best features in the sets: with order 100 (top left) and with order 2000 (bottom left). In the right side, we see the correlation plots for Clustering and with xmin standardization for the sets: with order 100 (top right) and with order 2000 (bottom right). High definition versions of these plots and additional plots for each top features for every order sets or standardization are available online at [18].**

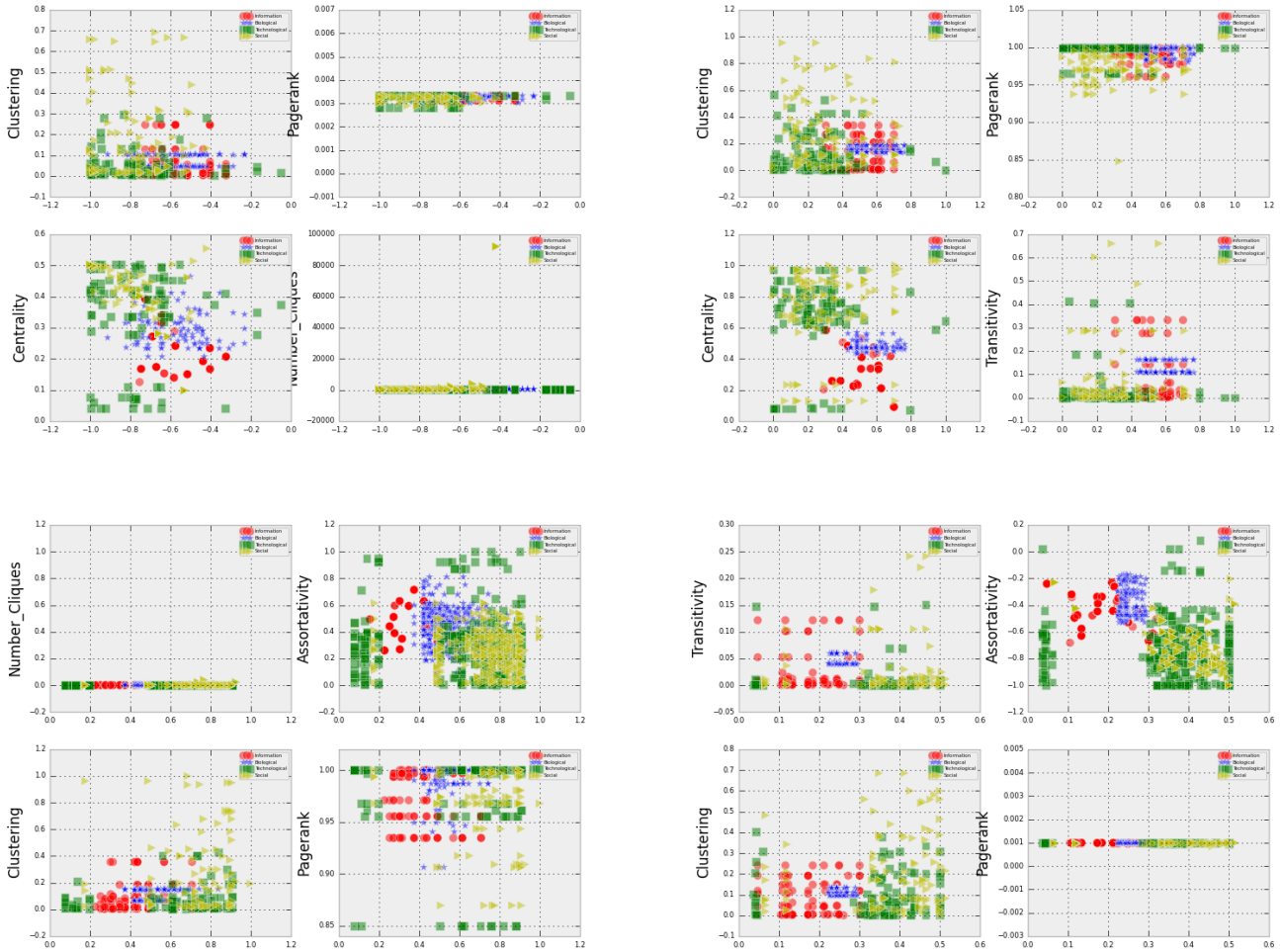

Figure 3: In the left side we see correlation plots for Assortativity, with no standardization, for the for best features in the sets: with order 300 (top left) and with order 1000 (bottom left). In the right side, we see the correlation plots for Clustering and with xmin standardization for the sets: with order 300 (top right) and with order 1000 (bottom right). High definition versions of these plots and additional plots for each top features for every order sets or standardization are available online at [18].

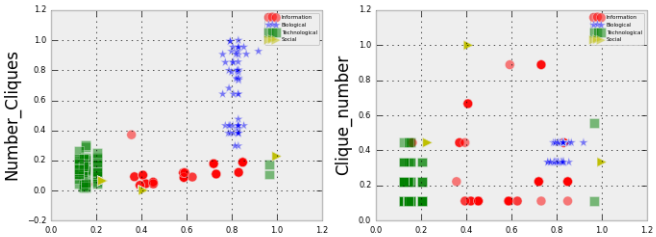

Figure 4: **Correlation plots for Centrality for set order 1000, without outliers, xmin standardization, for the best scoring features for this set and xmin normalization.**

belonging to the opposite label. For large networks (orders of 1000 and 2000), we obtain 100% testing accuracy for any of the four types of networks (bio, tech, info, and social) for any of the three types of standardization (xmin, gauss, none). In addition, all of the other sets show accuracy of at least 92%.

Logistic Regression

To be able to identify the best features for network classification, we use pairwise **logistic regression** classification. The features selected more often are selected as good features. For each of five sampling sets and for each of the standardization methods, we were able to select which of the 19 features[1] that are more relevant for the classification. We first analyze the sets with outliers and then without outliers[2]. When including outliers, we achieve an average of than 95% testing accuracy for the large sets (order 1000 and 2000), independently of the sets being standardized or not. An example of correlation plot for sets without outliers with xmin standardization is shown, in the Fig. 4.

The worst performance was for the sets with order 300, without outliers and without standardization, achieving average of 73% testing set accuracy. Followed by the smallest sets, order 100, achieving an average of $88-90\%$ accuracy for any standardization/outlier case. The better performance of the 100 set over the 300 set can explained by the fact that many biological networks (metabolic type) and ego-centered networks (retweet) contain in general in the order of 100 nodes, only being present in the first sampling set. The worse performance of these sets among the larger sets can be explained by the fact that most of the networks are in fact larger than 500 nodes.

The best features for each case together with their performance over the testing sets can be seen in the table 4. The complete result can be seen in the table 5 (including outliers) and 5 (without outliers). Although the top scoring features slightly change depending on the order of the graph, we find that the most important features for classification are **clustering, pagerank, centrality, transitivity, number of cliques**, and **assortativity**. In addition, **number of triangles** and **square clustering** tend to be a good classification feature for smaller networks, while **assortativity** and **transitivity** tend to be more relevant for larger networks. We also see that the optimal number of features in this classification is 5 (Fig. 5).

[1]Since we divided our data a by five sampling orders, we do not use order as a classification feature.

[2]Outliers were removed case by case, by searching for very extreme points.

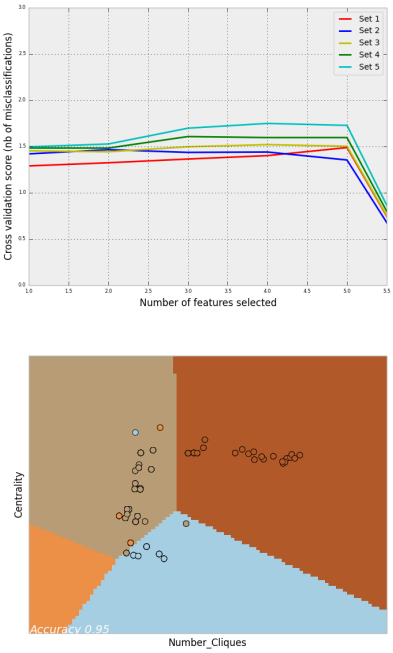

Figure 5: **(top) Optimal number of features for logistic regression classification, calculated from k-fold cross-validation. (bottom) An example of the separation surface for logistic regression for the set order 2000, one-vs-one.**

Support Vector Machine

Using **one vs. one SVM** classification, we report similar results from logistic regression for the best classification features. However, the performance of the former classifier seems to be slightly worse: we only find testing set accuracies above 80% for larger sets, as we can see in the table 4. The best performance results were for the Gaussian standardization. Sets without outliers did not performed better than 70% accuracy for non-standardized graphs and around 80% for Gaussian standardized graphs. This shows that the outliers might carry important characteristics of these networks, but it can also be due to the great weight that theses points add to the learning task. This results are corroborated when we report 99-100% testing set accuracy for set order 2000, with zeros and with outliers. Examples of the separation surface for several classifiers for a set with outliers is shown in the Fig. 6.

Unsupervised Learning

As a complementary analysis we attempt to classify our networks using the **k-means** algorithms for 4 cluster. The results failed to present meaningful classification, with very few examples that partially worked. In the Fig. 7 we see one of them, for the set of order 100.

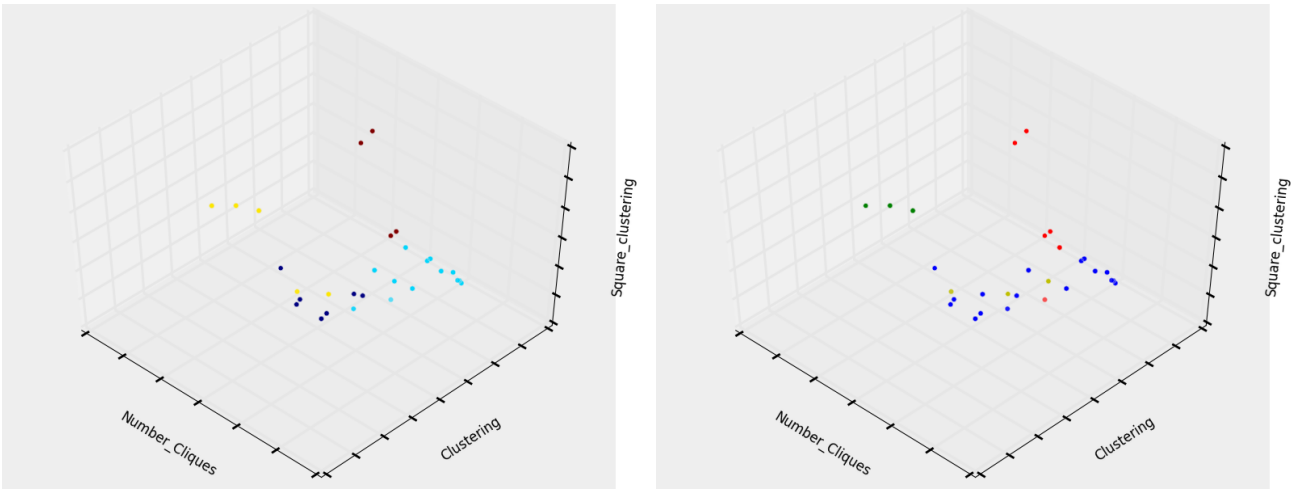

Figure 7: **K-means clustering for 4 classes, in terms of three features: number of cliques, clustering, and square clustering, for the xmin standardization (left). Ground plot (no classifier, just the date). We see that the cluster were close but not completely right. All the other plots are available at [18].**

Stand.	Set	O?	Best Features	Acc.
xmin	100	F	NCl, **Clu**, Scl, **Pag**, **Cen**	0.9
xmin	300	T	Den, Scl, Com, **Pag**	0.92
xmin	300	F	**Ass**, **Clu**, **Pag**, **Cen**	0.85
xmin	500	T	**Tra**, NTr, Scl, **Pag**	0.97
xmin	500	F	**Tra**, **Clu**, Den, Scl, **Pag**, **Cen**	0.92
xmin	1000	T	Cnu, **Cen**	0.98
xmin	1000	F	**Ass**, **Clu**, **Pag**, **Cen**	0.93
xmin	2000	T	Cor	0.98
xmin	2000	F	**NCl**, **Ass**, **Tra**, **Clu**, **Pag**, **Cen**	0.94
gaus	100	T	NCl, **Clu**, Scl, **Pag**, **Cen**	0.86
gaus	100	F	NCl, **Clu**, Scl, **Pag**, **Cen**	0.9
gaus	300	T	Den, Scl, Com	0.93
gaus	300	F	**Clu**, Den, **Pag**, **Cen**	0.84
gaus	500	T	**Tra**, NTr, Scl, **Pag**	0.99
gaus	500	F	**Tra**, **Clu**, Scl, **Pag**, **Cen**	0.9
gaus	1000	T	NCl, Cnu, **Cen**	1
gaus	1000	F	**Ass**, **Clu**, **Pag**, **Cen**	0.93
gaus	2000	F	**NCl**, **Ass**, **Tra**, **Clu**, **Pag**, **Cen**	0.98
none	100	T	NCl, **Clu**, Scl, **Pag**, **Cen**	0.94
none	100	F	NCl, **Clu**, Scl, **Pag**, **Cen**	0.86
none	300	T	Den, Scl, Com, **Pag**	0.89
none	300	F	**Clu**, **Pag**, **Cen**	0.91
none	500	T	**Tra**, NTr, **Pag**	0.78
none	500	F	**Tra**, **Clu**, Den, Scl, **Pag**, **Cen**	0.98
none	1000	T	Cnu, **Cen**	0.98
none	1000	F	**Ass**, **Clu**, **Pag**, **Cen**	0.8
none	2000	T	Cor, **Cen**	0.98
none	2000	F	**NCl**, **Ass**, **Tra**, **Clu**, **Pag**, **Cen**	0.92

Table 1: **The features that best scored in logistic regression, with their respective overall testing accuracy. The first column is the type of standardization, the second column is the set order, the third column is T(rue) for sets containing outliers or F(alse) otherwise, and the last column is the average testing accuracy. We can see that the most important features for this classifier are clustering, pagerank, centrality, transitivity, clique number, and assortativity.**

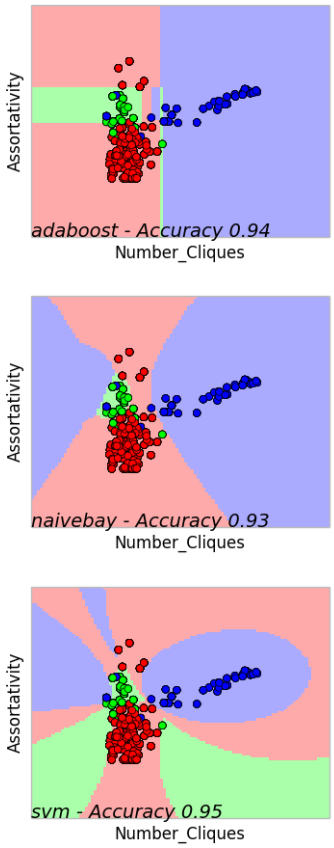

Figure 6: **Separation surface for the set order 2000, one-vs-one:(top) Adaboost (accuracy 0.94), (middle) Naive Bayes (accuracy 0.93) , and (bottom) SVM (0.95). This also illustrates that the separation surface is not linear. Note that the fourth cluster (biological) is ill represented in this example, and this is due the fact that most of the biological networks have less than 2000 nodes.**

Set	Stand.	Best Features	Test. Acc.
No outliers,	gauss	Clu, Cen	0.8
no zeros, set		NCl, Pag	0.82
order 2000		Tra, Pag	0.81
With outliers,	gauss	Cen, Tran	0.93
no zeros,		Cen, NCl	0.96
set order		Cen, Pag	0.93
1000		Cen, Clus	0.91
With outliers,	gauss	Cen, Tra	0.94
no zeros,		Cen, Clu	0.95
set order		Cen, NCl	0.97
2000		Tra, Pag	0.95
With outliers,	gauss	Cen, Tra	0.91
with zeros,		Cen, Clu	0.91
set order		Cen, NCl	0.96
1000		Cen, Pag	0.91
With outliers,	gauss	Cen, Tra	0.96
with zeros,		Cen, Clu	0.95
set order		Cen, NCli	0.95
2000		Tra, Pag	0.93

Table 2: The features that scored best in the multi-label one vs. one SVM classification of the networks, with their respective overall testing accuracy. We see that the most important features for this classifier are clustering, pagerank, centrality, transitivity, and number of cliques.

5. CONCLUSIONS

In this work we performed several learning analysis of an extensive set of real world complex networks. We outline some of the conclusions:

- **Best classification features**: The top three features are clustering, pagerank, and centrality. They are followed by transitivity, assortativity, and number of cliques.
- **Validity of the data sampling**: The sampling method developed here seems to have a good performance overall, only underestimating some of the properties for the small networks. This resulted in the set of order 300 having the wort performance between the five sets.
- **Validity of the data standardizing**: The standardizing sets performed similar than the xmin and gauss sets for logistic regression. For the SVM classifier, the standardized set performed better.
- **Validity of the classifiers**: We reported testing set accuracies larger than 90% for the supervised classifiers, for most of the data set configurations. Sets with outliers tend to report better accuracies. The unsupervised learning was not very successful and further studies are needed.
- **Some comments on the features**: Most of the networks are sparse, while social and technological tend to be general denser. These two networks also tend to present higher transitivity. However, large standard deviations shows the heterogeneity of the networks for this feature. For most networks, we observe most of the nodes having either very low or very high eccentricities. In terms of diameters, the order of magnitude of the diameter is the same for the most domains, with exception of social and biological networks. The same observation does not hold for the radius, which is roughly similar for most domains. Information networks seems to show a radius of hundreds of hops, instead of tens for the other domains. For most networks, the betweenness are homogeneous, following a normal-like distribution. The presence of only a few central links supports the hypothesis that the networks are modular. Some results available from the literature are exposed in the tables in final of this paper.

6. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. G. Li *Graph Classification via Topological and Label Attributes*

[2] H. Fei, J. Huan, *Structure Feature Selection For Graph Classification*

[3] Y. Keneshloo, S. Yazdani, *A Relative Feature Selection Algorithm for Graph Classification*

[4] B. Kantarci, V. Labatut, *Classification of Complex Networks Based on Topological Properties*

[5] J. Ugander et. al, *Subgraph Frequencies: Mapping the Empirical and Extremal Geography of Large Graph Collections*

[6] K. Goh et. al., *Classification of Scale-free Networks*

[7] SNAP Database, `http://snap.stanford.edu/data`

[8] KONECT Database `http://konect.uni-koblenz.de/networks`

[9] ND database. http://www3.nd.edu/ networks/resources/metabolic/index.html

[10] VLADO database. http://vlado.fmf.uni-lj.si/pub/networks/data

[11] M. E. J. Newman, *Networks: An Introduction*, Oxford University Press (2010).

[12] M. E. J. Newman, *Mixing patterns in networks*, Phys. Rev. E 67, 026126 (2003)

[13] M. E. J. Newman; *The Structure and Function of Complex Networks*

[14] Leskovec, Jure and Faloutsos, Christos. *Sampling from large graphs* ACM SIGKDD 2006.

[15] Sang Hoon Lee, Pan-Jun Kim, Hawoong Jeong *Statistical properties of sampled networks* Phys. Rev. E 73, 016102 (2006)

[16] Minas Gjoka, Maciej Kurant, Carter T Butts, Athina Markopoulou. *Walking in Facebook: A Case Study of Unbiased Sampling of SNs*

[17] http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html

[18] http://astro.sunysb.edu/steinkirch/new/mloutputs.html

[19] https://github.com/mariwahl/NetAna-Complex-Network-Analysis

[20] https://github.com/mariwahl/NetClean-Complex-Networks-Data-Cleanser

[21] https://github.com/mariwahl/MLNet-Classifying-Complex-Networks

Stand.	Set	Ord	Siz	Ass	Tra	Deg	Cor	NTr	NCl	Cnu	Clu	Eco	Ecc	Dia	Bet	Den	Rad	Scl	Com	Pag	Cen
xmin	100	0.3	0.5	0.8	0.4	0.4	0.0	0.6	**1.0**	0.5	**1.0**	0.0	0.2	0.2	0.2	0.8	0.2	**1.0**	0.2	**1.0**	**1.0**
xmin	300	0.4	0.6	0.1	0.9	0.8	0.7	0.0	0.8	0.5	0.9	0.0	0.3	0.3	0.4	**1.0**	0.1	**1.0**	1.0	0.9	0.8
xmin	500	0.5	0.2	0.2	**1.0**	0.6	0.1	**1.0**	0.8	0.4	0.3	0.0	0.5	0.3	0.5	0.6	0.1	**1.0**	0.4	**1.0**	0.8
xmin	1000	0.4	0.1	0.1	0.2	0.6	0.8	0.2	0.9	**1.0**	0.1	0.0	0.4	0.2	0.2	0.4	0.1	0.1	0.6	0.7	**1.0**
xmin	2000	0.1	0.0	0.0	0.2	0.4	0.9	0.0	0.3	0.4	0.0	0.0	0.2	0.2	0.0	0.6	0.0	0.0	0.2	0.3	**1.0**
gaus	100	0.4	0.6	0.8	0.3	0.4	0.0	0.7	**1.0**	0.5	**1.0**	0.0	0.3	0.1	0.3	0.7	0.2	**1.0**	0.1	**1.0**	**1.0**
gaus	300	0.5	0.5	0.1	0.8	0.8	0.7	0.1	0.7	0.4	0.9	0.0	0.2	0.4	0.3	**1.0**	0.1	**1.0**	1.0	0.9	0.9
gaus	500	0.5	0.2	0.2	**1.0**	0.6	0.1	**1.0**	0.8	0.4	0.3	0.0	0.5	0.2	0.5	0.6	0.1	**1.0**	0.4	**1.0**	0.8
gaus	1000	0.4	0.2	0.1	0.2	0.6	0.8	0.3	0.9	**1.0**	0.1	0.0	0.4	0.2	0.2	0.3	0.1	0.1	0.7	0.7	**1.0**
gaus	2000	0.1	0.0	0.0	0.2	0.4	0.9	0.0	0.2	0.5	0.0	0.0	0.2	0.2	0.1	0.6	0.0	0.0	0.2	0.4	**1.0**
none	100	0.3	0.6	0.9	0.4	0.5	0.0	0.6	**1.0**	0.4	**1.0**	0.0	0.3	0.1	0.2	0.8	0.2	**1.0**	0.1	**1.0**	**1.0**
none	300	0.5	0.5	0.2	0.8	0.8	0.7	0.1	0.8	0.4	0.9	0.0	0.2	0.3	0.4	**1.0**	0.1	**1.0**	1.0	0.9	0.9
none	500	0.4	0.2	0.3	**1.0**	0.6	0.1	**1.0**	0.8	0.4	0.3	0.0	0.5	0.3	0.5	0.6	0.1	**1.0**	0.5	**1.0**	0.7
none	1000	0.5	0.1	0.1	0.2	0.6	0.7	0.8	0.8	**1.0**	0.1	0.0	0.3	0.2	0.2	0.4	0.1	0.1	0.6	0.6	**1.0**
none	2000	0.1	0.0	0.0	0.2	0.4	0.9	0.0	0.3	0.3	0.1	0.0	0.2	0.2	0.0	0.6	0.1	0.0	0.3	0.4	**1.0**

Table 3: **Feature analysis for multilabel logistic regression, one vs. all, for each order set number and standardization type, including outliers. The bold values are the best features.**

Stand.	Set	Ord	Siz	Ass	Tra	Deg	Cor	NTr	NCl	Cnu	Clu	Eco	Ecc	Dia	Bet	Den	Rad	Scl	Com	Pag	Cen
xmin	100	0.3	0.7	0.8	0.4	0.4	0.1	0.6	**1.0**	0.5	**1.0**	0.0	0.3	0.1	0.2	0.8	0.2	**1.0**	0.6	**1.0**	**1.0**
xmin	300	0.3	0.8	**1.0**	0.6	0.2	0.1	0.0	0.7	0.5	**1.0**	0.0	0.4	0.2	0.5	0.9	0.5	0.1	0.0	**1.0**	**1.0**
xmin	500	0.2	**1.0**	0.7	**1.0**	0.3	0.0	0.7	0.8	0.8	**1.0**	0.0	0.1	0.1	0.1	0.9	0.3	**1.0**	0.0	**1.0**	**1.0**
xmin	1000	0.6	0.4	**1.0**	0.8	0.2	0.3	0.3	0.6	0.6	**1.0**	0.0	0.5	0.6	0.6	0.1	0.2	0.5	0.0	**1.0**	**1.0**
xmin	2000	0.2	**1.0**	**1.0**	**1.0**	0.5	0.6	0.4	0.7	0.0	**1.0**	0.0	0.4	0.3	0.3	0.7	0.1	0.8	0.0	**1.0**	**1.0**
gaus	100	0.3	0.7	0.7	0.3	0.3	0.1	0.5	**1.0**	0.4	**1.0**	0.0	0.3	0.1	0.2	0.8	0.3	**1.0**	0.5	**1.0**	**1.0**
gaus	300	0.2	0.9	**1.0**	0.6	0.2	0.0	0.0	0.8	0.4	**1.0**	0.0	0.4	0.2	0.4	0.9	0.5	0.2	0.0	**1.0**	**1.0**
gaus	500	0.3	**1.0**	0.7	**1.0**	0.3	0.0	0.0	080	0.7	**1.0**	0.0	0.1	0.1	0.1	0.9	0.3	**1.0**	0.0	**1.0**	**1.0**
gaus	1000	0.6	0.5	**1.0**	0.7	0.2	0.3	0.2	0.8	0.6	**1.0**	0.0	0.5	0.6	0.5	0.1	0.2	0.5	0.0	**1.0**	**1.0**
gaus	2000	0.3	**1.0**	**1.0**	**1.0**	0.5	0.6	0.3	0.8	0.0	**1.0**	0.0	0.5	0.3	0.2	0.7	0.0	0.8	0.0	**1.0**	**1.0**
none	100	0.3	0.6	0.8	0.3	0.4	0.0	0.5	**1.0**	0.5	**1.0**	0.0	0.2	0.1	0.2	0.8	0.2	**1.0**	0.5	**1.0**	**1.0**
none	300	0.3	0.9	**1.0**	0.6	0.2	0.1	0.0	0.5	0.4	**1.0**	0.0	0.3	0.2	0.5	0.9	0.5	0.3	0.0	**1.0**	**1.0**
none	500	0.2	**1.0**	0.7	**1.0**	0.3	0.0	0.0	0.8	0.7	**1.0**	0.0	0.1	0.1	0.1	0.9	0.3	**1.0**	0.0	**1.0**	**1.0**
none	1000	0.7	0.5	**1.0**	0.8	0.1	0.3	0.2	0.8	0.7	**1.0**	0.0	0.6	0.6	0.5	0.1	0.2	0.4	0.0	**1.0**	**1.0**
none	2000	0.3	**1.0**	**1.0**	**1.0**	0.5	0.6	0.4	**0.9**	0.0	**1.0**	0.0	0.4	0.2	0.3	0.7	0.0	0.9	0.0	**1.0**	**1.0**

Table 4: **Feature analysis for multilabel logistic regression, one vs. all, for each order set number and standardization type, without outliers. The bold values are the best features.**