MODEL SELECTION FOR SOCIAL NETWORKS USING GRAPHLETS TECHNICAL REPORT

MATTHEW S. HURSHMAN, J. JANSSEN, AND NAUZER KALYANIWALLA

Abstract. Several network models have been proposed to explain the link structure observed in online social networks. Such models are usually justified by showing that they can replicate certain observed features of real networks, such as a power law degree distribution, a high degree of clustering, and the small world property. This paper addresses the problem of choosing the model that best fits a given real world network. We implement a model selection method based on un-supervised learning. An alternating decision tree is trained using synthetic graphs generated according to each of the models under consideration. Graphs are represented as feature vectors incorporating the frequency counts of all connected subgraphs on 3 and 4 vertices as well as features describing the degree distribution and small world property. We show that the subgraph counts alone are sufficient in separating the training data. For the test data, we take four Facebook graphs from various American Universities. Our results show that models incorporating some form of the preferential attachment mechanism tend to perform the best.

1. Introduction

Over the last couple of decades, the study of social networks has revealed many distinguishing features of social networks such as a power law degree distribution, high clustering coefficients and small hop distances between individuals [13]. A number of models have been developed that can replicate these features. The ability of these models to replicate some of the observed features has been historically the sole justification to conclude that the model is an appropriate one for social networks. However, two graphs with similar global features can have drastically different local structure. Whether or not models can effectively generate similar local structure has been widely ignored when proposing models for real networks. Recent work in validating models for biological networks [11], [12], has focused on the distribution of small subgraphs in the network, called the graphlets, thus pushing the local structure of the network to the forefront of the validation process. Work in graph similarity methods [16],[17], have also begun focusing on comparing networks using graphlets.

The goal of this work is to find which of six selected models is the most capable of replicating a real online social network. Finding the network model most appropriate for online social networks provides insight into the growth mechanism that is the most dominant in determining the structure of online social networks. We consider models based on preferential attachment, copying and embedding the nodes in a geometric space. A specific question addressed is whether a *geometric model* is appropriate. In a geometric model, nodes are assumed to be embedded in a metric space, and formation of edges is influenced by the metric distance between vertices. (For a recent overview of geometric models, see [8]). The advantage of a geometric model is that the metric space can be seen as the *social space* representing the interests, hobbies and other attributes of the individuals corresponding to the nodes of the social network. Thus, assuming a geometric model gives the possibility for inference of the social space from the network, thus providing a basis for identifying communities or individuals with similar interests.

Our approach to model validation is largely adopted from work by Middendorf *et al.* [11] for validating models for protein-protein interaction networks via discriminative classification techniques from machine learning. In our work we modify their approach and extend it to online social networks. Our work is different in two ways: (i) we build our decision tree using a different algorithm, and (ii) we extend their approach to much larger and denser graphs.

We build an alternating decision tree classifier using the LADTree algorithm [22]. We use graphs generated from our proposed models as training data and snapshots of the online social network Facebook as our test data. Our Facebook data comes from the Mason Porter Facebook 100 dataset and was downloaded from [15]. We represent the Facebook graphs as feature vectors. Features included describe both the local and global structure of the graphs. To represent the local structure, we include the counts of various small connected subgraphs called *graphlets* as features. To represent global structure, we include features which describe the degree distribution and the small world property. Specifically, we include percentiles of the degree distribution, the assortativity coefficient [18] and the average path length as features.

We verify the conclusion of [11] that graphlets can effectively separate different models and we show that including information about global features does not greatly improve the classification accuracy on the training models.

In the next section we describe our method in detail. We present, descriptions of the models used, the testing data used (Facebook), the features selected to represent the graphs and the classification algorithm selected. In Section 3 we discuss the results of our experiment. We first study the performance of our classifier using a test set containing graphs generated from our training models. We conclude, that graphlet counts work extremely well for separating the different models. We also observe that including features corresponding to the global structure of the graph does not increases the classification accuracy. We also test how robust the classifier is by artificially creating noise in the data by changing some of the edges. Finally, we evaluate how well our models perform when we use our Facebook data as test data. Though we only test four different Facebook graphs, we observe that models which incorporate some level of preferential attachment tend to perform the best.

2. Experimental Procedure and Implementation

Our model selection method follows three steps. First, we obtain the training data, by generating 1000 graphs according to each of the six models we have selected: the Preferential Attachment Model, the Copy Model, the Random Geometric Model (2D and 3D), and the Spatial Preferred Attachment Model (2D and 3D). The details of the models are given in Section 2.1 below. The parameters of the models are randomly sampled from a range such that the graphs generated are similar in size and density to the test data. The restriction of the sample range of the parameters is necessitated by the fact that the graphlet counts depend heavily on the size and density of a graph, even for graphs generated by the same model. For this reason it is necessary to generate a new training set for each test graph which greatly increases the amount of time to test different Facebook graphs. It is currently unclear whether there exists an adequate normalization method, which would make it possible to compare graphlets counts for graphs of different densities.

Next, we use the training data to build a multi-class alternating decision tree (ADT). The details of the construction of the ADT are given in Section 2.3 below. We represent the graphs using features that capture both the local structure of the graph, through the graphlets, and the global structure. A description of the features is given in Section 2.2 below.

Finally, we compute the feature vectors corresponding to on-line social network data, in this case snapshots of Facebook. Running this feature vector through the classifier gives a score for each model corresponding to how well the model fits the test data. Our experimental procedure is repeated for four different Facebook networks taking from the following American universities : Princeton, American University, MIT and Brown. We obtained this data from Mason Porter's et al. Facebook100 data set which is introduced in[27]. We discuss the results of these four experiments in Section 3.

2.1. **Models.**

We have implemented six different graph models. Our choice of models was motivated by the desire to test a wide range of models commonly proposed for social networks, based on a number of different attachment principles. Special attention was given to spatial models, a class of models that is gaining support because of the ability to model node attributes through spatial representation. Wherever more than one variation of the model has been proposed in the literature, we have opted for the simpler versions. This choice was motivated by the wish to avoid ambiguity in the classification.

All model generation algorithms are written in Python using the graph-tool module [14]. Our training set includes only undirected graphs without multiple edges. In the case where a model generates a directed graph, we ignore the direction of the edges after generation and remove any multiple edges.

Preferential Attachment Model (PA). The Preferential Attachment model was first introduced by Barabási and Albert in [3] as a model for the World Wide Web. The model incorporates the mechanism that a new user is more likely to attach to a user that is "well known" (has many incident edges). The model has two parameters, $d \in \mathbb{N}$ and $\alpha \in [0, d]$. It generates a sequence of graphs $\{G_t : t \in \mathbb{N} \cup \{0\}\}$ where $G_0 = H$ is a small random seed graph. We will take H to be the Erdös-Rényi random graph $G(100, ed)$, where ed is the edge density of the test graph. At each time step $t > 0$, G_t is formed by adding a new vertex v_t and adding d edges (v_t, w_i) where w_i is chosen with probability proportional to its degree. Precisely, the probability that w_i is chosen is given by:

$$P(w = w_i) = \frac{deg_{G_{t-1}}(w) + \alpha}{2d(t-1) + |E(H)| + \alpha(t-1)},$$

where $deg_{G_{t-1}}(w)$ is the degree of w in G_{t-1}. Thus, vertices of high degree are more likely to accumulate more edges.

Copy Model (COPY). The Copy Model was originally proposed in [4] as a model for the World Wide Web and further studied in other works such as [5],[6]. The idea behind this model is that a person tends to meet friends through a currently existing friend. The model has two parameters $p \in (0, 1)$ and $d \in \mathbb{N}^+$. It generates a sequence of directed graphs $\{G_t : t \in \mathbb{N} \cup \{0\}\}$. Again, $G_0 = H$ is a small random seed graph. We take H to be the directed version of the Erdös-Rényi random graph with 100 vertices and p equal to the edge density of the test graph. At each time step $t > 0$, we add a new vertex v_t. We then uniformly at random choose a vertex w from G_{t-1} and add an edge from v_t to each out-neighbour of w with probability p. We then chose d more vertices u uniformly at random and add directed edges (v_t, u). Since we are only interested in undirected graphs, we ignore edge orientation after we generate the graph.

Random Geometric Model (GEO). The Random Geometric Model is a model where the vertices are embedded in a geometric space and edges are determined by a threshold on the metric distance between two vertices. A theoretical analysis of a variety of the model corresponding to the case $p = 1$ in our description, can be found in [9]. The model was used to model protein-protein interaction networks in [12]. The model has three parameters, $r \in (0, 1)$, $p \in (0, 1)$ and $d \in \mathbb{N}^+$. In a geometric model, the vertices of the graph are uniformly at random embedded into a metric space S. If the distance between vertices in this space is less than a specified threshold r, then an edge is added with probability p. For small values of r, the GEO model has a large average path length value since short cuts between vertices that are far away are not possible. In other words, this model does not exhibit the small-world property. To this end we add a small number of random edges d so that some edges between distant vertices are possible. We consider a two dimensional (GEO2D) and a three dimensional (GEO3D) version of this model where the metric space S is $[0, 1]^2$ and $[0, 1]^3$ respectively, equipped with the torus metric.

Spatial Preferential Attachment Model (SPA). The Spatial Preferential Attachment model introduced in [7] combines a geometric model with a preferential attachment model. The vertices are placed in the same metric spaces as the two GEO models above giving us a two dimensional (SPA2D) and three dimensional (SPA3D) version of the SPA model. The model has three parameters $A_1 \in (0, 1)$, $A_2 \geq 0$, and $p \in (0, 1]$. We form a sequence of directed graphs $\{G_t\}$, $t \in \mathbb{N} \cup \{0\}$ with G_0 as the empty graph. We define a region of influence around a vertex

v at time $t \geq 1$, written $R(v,t)$, with area

$$|R(v,t)| = \frac{A_1 deg^-(v,t) + A_2}{t}$$

or $R(v,t) = S$ if the above is greater than 1. In the above, $deg^-(v,t)$ is the in-degree of v at time t. At each time step $t \geq 1$, a point in S is uniformly at randomly chosen to be the new vertex v_t. For each vertex $u \in V_{G_{t-1}}$ such that $v \in R(u, t-1)$, we independently add an edge from v_{t-1} to u with probability p. In this model, the influence regions are proportional to the in-degree of the vertex but decrease over time. Again, after model generation, we ignore the direction of the edges.

2.2. **Features.** We represent our graphs by 17 features in a vector representation. These features include information about the global properties of the graphs, specifically the degree distribution, the assortativity coefficient and the average path length between vertices. In addition, we capture the local structure through the raw graphlets counts for the connected subgraphs of size 3 and size 4. Below is a description of each of the features.

Degree Distribution Percentiles. The degree distribution is a favourite property studied for most real world networks. A distribution with a power law tail is a distinguishing property of many such networks, including online social networks. The most logical feature to use here would be the coefficient of the power law degree distribution. Unfortunately, not all the models generate graphs with a power law degree distribution (*e.g.* random geometric models). Also, even if a model does generate a power law degree distribution it can be difficult to determine its power law coefficient. Instead, to measure the spread of the degree distribution, we consider the percentiles of the distribution formed by breaking it evenly into 8 different pieces. This give us 7 features, called $deg_1, deg_2, deg_3, deg_4, deg_5, deg_6$, and deg_7.

Assortativity Coefficient. The assortativity coefficient $r \in [-1, 1]$ is a measurement of how well vertices of similar degree link to one another in the network. An assortativity coefficient close to -1 indicates that vertices tend to link to vertices of different degrees and a value close to 1 indicates that vertices tend to link to vertices of similar degrees. It is shown in [13] that online social networks have positive assortativity coefficients while the World Wide Web and biological networks have negative assortativity coefficients. We compute the assortativity coefficient in graph-tool using the following equation from [18],

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}$$

where e_{ij} is the fraction of edges from a vertex of degree i to a vertex of degree j and $a_i = \sum_j e_{ij}$ and $b_j = \sum_i e_{ij}$.

Average Path Length. The small world property, implying a small average distance between nodes, is another distinguishing aspect of social networks. It is shown in [13] that online social networks have small average path length. Here we will compute the average path length between nodes by selecting 100 random pairs of nodes and calculate the length of the shortest path between them using a breadth-first-search which is implemented in graph-tool.

Graphlets. To characterize local structure, we include as features all the counts of connected subgraphs of size 3 (two nonisomorphic graphs) and 4 (six nonisomorphic graphs), as shown in Figure 1 below. Unfortunately, no algorithm is known which computes the full counts for these subgraphs efficiently. As a compromise, we use the sampling algorithm of Wernicke [19] to sample the number of these graphlets. As input, Wernicke's algorithm takes in a labeled graph and an integer k, the size of subgraphs to be counted. The algorithm generates a tree of depth k by looping through each vertex and performing a depth first search on the k-neighbourhoods of each vertex. When the algorithm terminates, a tree of depth k has been formed where the leaves of the tree correspond to all the size k connected subgraphs of G. Building the entire tree is extremely time consuming for the size of graphs we are considering. Wernicke proposes a sampling procedure using his algorithm. A specified proportion p of the subgraphs can be

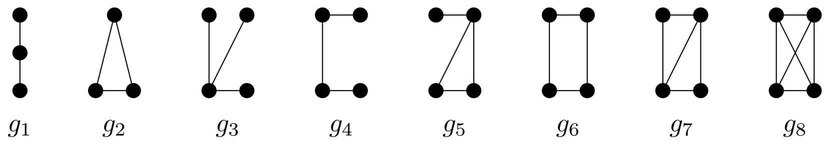

FIGURE 1. The graphlet features

found by skipping steps in the algorithm with a probability which depends on the depth of the tree. Experiments performed by Wernicke in [19] show that the sampling algorithm samples the correct proportion.

For our experiments, we sample 1% of the size 3 graphlets and 0.01% of the size 4 graphlets. The computation of the graphlets is by far the most time consuming of all the features. For the size and density of graphs we are considering it was not feasible to include subgraphs of size greater than 4. In [11], the authors consider subgraphs up to size 7 but this is only possible because the graphs are much smaller and sparser then those considered here. Inclusion of graphlets of larger size will only be possible for graphs of the size and density we consider here if new methods are developed to compute or estimate graphlets counts which show a dramatic increase in efficiency.

2.3. **Classification.**

To classify our data we use the multi-class alternating decision tree (ADT) algorithm LADTree of Holmes *et al.* [22]. ADTs are a class of boosted decision trees which were introduced by Freund and Mason in [20]. Boosting [21] is a well established classification technique which combines so called "weak classifiers" to form a single powerful classifier. In successive steps called *boosting steps*, weighted combinations of the weak classifiers are applied to the training data, and the weights are adjusted in each step to improve the classification.

The first ADTs were built using the AdaBoost boosting algorithm [21]. The ADT used here, LADTree, is built on the lesser known LogitBoost boosting algorithm of Friedman, Hastie and Tibshirani [23]. Friedman *et al.* show in their work that both boosting algorithms are fitting an additive logistic regression model. They argue that LogitBoost is the more appropriate algorithm, because it fits the regression model using the more typical maximum likelihood minimization criteria, whereas AdaBoost uses an exponential minimization criteria.

In Figure 2, we show a partial LADTree which was constructed during our experiment. An ADT has two types of nodes, *decision nodes* (rectangles in Figure 2) and *prediction nodes* (ellipses in Figure 2). Decision nodes contain a boolean predicate which corresponds to a threshold on one of the features in the feature vectors for the training data. The prediction nodes contain real-valued scores, one for each of the classes in the training set. In our case, we have six different classes or models so each prediction node contains six scores.

The LADTree begins with a prediction node which has a score of zero for each of the models. In each boosting iteration, a decision node is added to the tree along with two prediction nodes as its children in the tree. The new decision node can be added as a child to any existing prediction node in the tree. The placement of the decision node and its Boolean predicate is the one that gives the best separation of the training data. The exact criteria for this is provided by the LogitBoost algorithm [23].

Once the LADTree has been formed, new instances, typically called the test data, can be classified by the tree. For us, the test data is the feature vector for the Facebook graph we wish to classify. The feature vector for the Facebook graph will determine its flow through the tree. The test instance travels through all possible paths it can reach in the tree resulting in a classification score which is the sum of all prediction nodes reached along the way. This results in six different scores, one for each of the six different models, F_j, $j = 1, 2, 3, 4, 5, 6$. A positive score is a good fit, a negative score is a bad fit. The model which obtains the highest score is

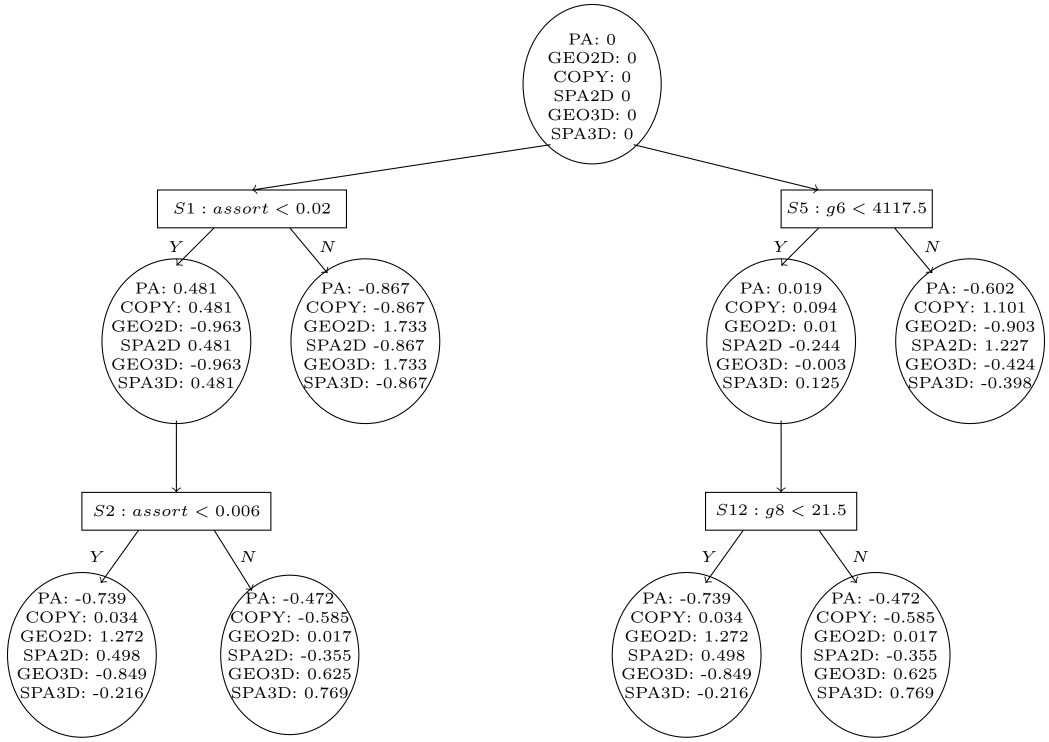

Figure 2. Partial LADTree using the full feature vector with 200 boosting iterations.

deemed to be the model that best describes the test data. The absolute values of the scores provide the level of confidence in the prediction. Thus, a large positive F_j indicates that model j is a good model for the test instance and a large negative F_j indicates that model j is a bad model for the test instance. The scores F_j can be readily interpreted as class probabilities p_j by the equation $p_j = \frac{e^{F_j}}{\sum_{j=1}^{6} e^{F_j}}$ which results in inverting the additive logistic model which is fitted by the LADTree algorithm [22].

The advantage of using ADTs is that they require no specific assumption about the geometry of the input space for the features. Thus we are free to incorporate any range of features such as degree distribution percentiles, average path length and subgraph counts without considering any potential dependence amongst them. The importance of each feature is based on how well it separates the 6 different models. We use the Weka software package for Java [25] to train all the LADTrees used in our experiments.

3. Results

We tested our approach on four different social network graphs taking from Mason Porter's et al. Facebook100 data set downloaded from [15]. Each graph in the data set corresponds to users at different universities. For our test data we take: Princeton University which has 6596 vertices and 293329 edges, American University which has 6386 vertices and 217661 edges, MIT which has 6440 vertices and 251252 edges and Brown University which has 8600 vertices and 384525 edges. In these graphs, each vertex corresponds to a Facebook user, and two vertices are connected if they are Facebook "friends".

For each of these graphs, the process is as follows. First, we generate a training set of 6000 graphs which are the of same size as the Facebook graph, and have edge density which differs by at most 5% from that of the Facebook graph. In order to test the effect of different features and a different number of boosting iterations, we build 9 LADTree classifiers. The classifiers are built using 3 different types of feature vectors; the *full feature vector* which incorporates all 17 features described in Section 2.2, the *graph feature vector* which uses only the graphlet features and the *other feature vector* which uses only the non-graphlet based features. For each of the feature vectors under consideration we build a classifier using 50, 100 and 200 boosting iterations, giving 9 classifiers in total for each experiment. To build the classifiers we use the well known machine learning software package Weka [25]. Finally, we use the classifiers to classify the Facebook graph. The model which produces the graphs which get the best score is considered to be the best model for the data.

3.1. **Testing the Classifier.** Before performing our experiments on the actual Facebook data, it is important to test the classifier to find out how we should interpret the results. To this end, we generate an additional 100 graphs for each of the models, and apply the classifier to this known data set. Since we know exactly which class these synthetic graphs belong to, this should establish an important baseline for the maximum and minimum possible scores achievable by each model.

We also test the robustness of the classifier. To do this, we take the 600 synthetic graphs and change a percentage of the edges by removing an edge from the graph and replacing it with a new edge chosen uniformly at random. The goal is to see how fast the classification accuracy deteriorates if an ever greater number of edges are changed. Overall, we have 6 test data sets of 600 graphs each, with 0%, 5%, 10%, 15% , 20% and 25% of the edges randomly changed. We generate the initial 600 graphs with the same density as the Princeton network and classify them using the LADTree classifiers we have generated for the Princeton data. To determine the importance of the graphlet features, we consider the classifiers built using both the full feature vector and the graph feature vector.

Models	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
PA	8.96 ± 1.18	-3.91 ± 2.39	-4.16 ± 1.18	0.17 ± 1.69	-2.38 ± 1.25	1.32 ± 0.82
COPY	-2.3 ± 0.34	7.02 ± 0.24	-2.19 ± 0.27	-0.19 ± 0.23	-3.2 ± 0.28	0.85 ± 0.27
GEO2D	-6.78 ± 1.59	-7.82 ± 3.55	9.13 ± 2.89	2.65 ±2.42	3.57 ± 1.47	-0.76 ± 1.55
SPA2D	-5.51 ± 2.5	-11 ± 3.86	2.89 ± 2.27	10.16 ± 3.05	-2.36 ± 2.04	5.81 ± 1.76
GEO3D	-6.14 ± 1.31	-8.42 ± 3.18	3.58 ± 1.61	-0.73 ± 1.05	9.04 ± 2.94	2.67 ± 2.32
SPA3D	-4.09 ± 2.48	-9.97 ± 4.54	0.03 ± 2.2	5.22 ± 2.06	-0.26 ± 2.79	9.07 ± 2.84

TABLE 1. Full Feature 50 Boosting iterations Average value with standard deviation

Models	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
PA	11.92 ± 0.89	-4.61 ± 1.25	-4.61 ± 1.93	2.39 ± 1.65	-5.11 ± 1.46	0.03 ± 1.03
COPY	-5.5 ± 1.67	11.73 ± 0.80	-0.34 ± 1.11	1.37 ± 1.02	-8.25 ± 1.93	0.99 ± 1.4
GEO2D	-10.83 ± 1.96	-10.64 ± 4.54	12.59 ± 3.19	4.07 ± 2.42	6.02 ± 1.91	-1.2 ± 1.84
SPA2D	-8.08 ± 3.57	-13.61 ± 4.57	3.04 ± 2.88	13.79 ± 3.72	-2.38 ± 3.45	7.25 ± 1.99
GEO3D	-10.72 ± 2.21	-12.55 ± 5.11	5.81 ± 2.30	1.56 ± 2.07	13.25 ± 3.79	2.66 ± 2.4
SPA3D	-6.62 ± 3.79	-13.04 ± 5.68	-0.09 ± 2.97	6.94 ± 2.17	-0.45 ± 4.13	13.26 ± 4.05

TABLE 2. Full Feature 100 Boosting iterations Average value with standard deviation

Models	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
PA	16.69 ± 1.5	-5.9 ± 1.55	-6.62 ± 2.75	2.42 ± 2.21	-8.90 ± 2.15	2.31 ± 1.66
COPY	-8.2 ± 4.9	18.31 ± 0.8	-6.31 ± 1.61	1.95 ± 2.03	-9 ± 2.46	3.24 ± 2.05
GEO2D	-16.79 ± 4.09	-18.23 ± 7.03	19.27 ± 3.01	5.48 ± 3.05	9.41 ± 3.44	0.86 ± 3.18
SPA2D	-10.75 ± 5.6	-20.41 ± 6.61	5.46 ± 4.24	19.53 ± 4.40	-3.71 ± 5.34	9.88 ± 2.56
GEO3D	-17.57 ± 4.34	-21.78 ± 8.46	8.92 ± 3.44	2.32 ± 2.96	20.5 ± 4.98	7.6 ± 4.12
SPA3D	-8.73 ± 5.76	-20.74 ± 7.99	0.82 ± 4.55	9.59 ± 2.7	0.07 ± 5.94	18.99 ± 4.06

TABLE 3. Full Feature 200 Boosting iterations Average value with standard deviation

First, consider the scores generated by the classifier for the unchanged synthetic graphs, shown in Tables 1, 2, 3, and 4. As expected, the graphs are overwhelmingly assigned to the class corresponding to the model that generated them. The scores range roughly between -10 and 10 for 50 boosting iterations, -15 and 15 for 100 boosting iterations and -25 and 25 for 200 boosting iterations for both the full and graph features. The performance of the classifier is consistent over the different number of boosting iterations.

Table 4 shows the performance on the synthetic graph when only the graph feature vector is used. Again, almost all graphs are classified correctly. Also, the synthetic graphs receive a similar or higher score for the correct class by using the graph feature vector, than by using the full feature vector. Thus we can conclude that graphlets alone are sufficient to recognize the graph structure of the models under consideration.

Models	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
PA	10.36 ± 0.51	0.45 ± 0.44	-5.26 ± 0.72	-0.6 ± 1.02	-6.19 ± 0.78	1.24 ± 0.98
COPY	0.07 ± 0.95	9.71 ± 0.85	-5.27 ± 0.36	0.35 ± 1.16	-5.95 ± 0.64	1.1 ± 0.36
GEO2D	-12.18 ± 3.11	-14.9 ± 4.77	13.86 ± 3.98	5.97 ± 3.11	6.53 ± 2.49	0.72 ± 1.85
SPA2D	-11.71 ± 2.89	-13.74 ± 4.32	2.35 ± 2.55	15.41 ± 3.44	-0.78 ± 3.31	8.47 ± 2.27
GEO3D	-13.21 ± 3.28	-15.36 ± 4.39	7.45 ± 1.86	3.27 ± 2.02	13.39 ± 2.84	4.46 ± 2.67
SPA3D	-11.41 ± 3.34	-14.22 ± 4.61	-0.03 ± 2.32	9.06 ± 2.17	1.62 ± 3.56	14.99 ± 3.03

TABLE 4. Graph Feature 100 Boosting iterations Average value with standard deviation

Next, we apply the classifiers to the modified graphs, to test the robustness of our classification method. Tables 5 and 6 give the classification accuracy for each of the 6 test data sets using the full feature vector and graph feature vector respectively. The classification accuracy on the original unchanged test data is very high for both the full and graph feature vectors. The classification accuracy is slightly but not significantly higher when only the graph feature is used. When 5% of the edges are changed the classification accuracy for the full feature drops to just below 75% while for the graph feature vector the accuracy is just below 80%. In this case, the graph feature vector alone performs significantly better than the full feature vector. For all other percentages of edge changes, the difference between the two is not significant. The conclusion of this experiment is that the graph features alone provide just as much information as the full feature set. In fact, as is the case when 5% of the edge were changed, including additional non-graph information can decrease the accuracy of the classifier. When 10% of the edges are changed both feature vectors give classification accuracies around 65% which is still a fair performance. When 15% of the edges are changed, the accuracy for both feature vectors drops to around 55%. At 20% and 25%, the accuracy dips below 50%. The accuracy at this level is not good but it is still much better than random guessing which would give the correct classification about 16.67% of the time.

Edge Changes	Boosting Iterations		
%	50	100	200
0	94.67	95.67	97.17
5	73.83	71.5	74.33
10	64	63.33	65.17
15	57.33	56.17	56.33
20	51.17	48.67	48.83
25	44.17	43	41.17

TABLE 5. Full Feature Classification Accuracy

Edge Changes	Boosting Iterations		
%	50	100	200
0	94.83	96.67	97.83
5	78.67	79.83	79.67
10	64	63.5	63.67
15	56.17	55.67	54.8
20	49.33	48	48.17
25	44	40.5	40.67

TABLE 6. Graph Feature Classification Accuracy

By observing Tables 7 and 8 (and Tables 26, 27 28 and 29 in Appendix B), we can see that the 3D (GEO3D and SPA3D) models are very robust against the changing of edges while their 2D (GEO2D and SPA2D) counterparts are not. Even after 5% of the edges are switched, roughly half of the 2D models are classified as their 3D counterparts. When 25% of the edges have been changed, only around 5% of the 2D models are classified correctly, with most of the graphs being classified as the 3D counterpart. Meanwhile, the 3D models maintain a good classification accuracy even when 25% of the edges are changed. Another interesting observation is that the COPY model is also somewhat robust against the changing of edges. Even with 5% of the edges switched, all the COPY graphs are classified correctly. The accuracy dips to around 95% occurs when 10% of the edges are switched. Even when 25% of the graph is changed, the classification accuracy stays within 50%–70%. The PA model on the other hand is not robust against the changing of edges. The classification is perfect when no changes are made but quickly decreases as edge changes start to accumulate. The accuracy is still around 80% when 10% of the edges are changed but drops to 50% accuracy when 15% of the edges are changed. After 25% of the edges have been changed the classification accuracy is an abysmal 1% – 2%.

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	93	0	7	0	GEO2D
	0	0	0	95	0	5	SPA2D
	0	0	9	0	91	0	GEO3D
	0	0	0	5	0	95	SPA3D
5%	89	7	0	1	0	3	PA
	0	100	0	0	0	0	COPY
	0	0	9	0	91	0	GEO2D
	0	0	0	36	0	64	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	1	0	99	SPA3D
10%	80	12	0	0	0	8	PA
	0	95	4	1	0	0	COPY
	0	0	6	0	94	0	GEO2D
	0	0	0	6	0	94	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	3	0	97	SPA3D
15%	51	47	0	0	0	2	PA
	0	86	6	8	0	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	3	0	97	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	0	2	0	98	SPA3D
20%	21	79	0	0	0	0	PA
	0	77	2	20	1	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	2	0	98	SPA2D
	0	0	8	0	92	0	GEO3D
	0	0	0	4	0	96	SPA3D
25%	1	98	0	0	0	1	PA
	0	62	1	36	1	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	2	0	98	SPA2D
	0	0	9	0	91	0	GEO3D
	0	0	0	2	0	98	SPA3D

Table 7. Full Feature Vector with 100 Boosting Iterations

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	92	2	6	0	GEO2D
	0	1	0	97	0	2	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	0	4	0	96	SPA3D
5%	88	2	0	0	0	10	PA
	0	100	0	0	0	0	COPY
	0	0	49	2	49	0	GEO2D
	0	0	0	47	0	53	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	1	0	99	SPA3D
10%	78	14	0	0	0	8	PA
	0	94	0	6	0	0	COPY
	0	0	11	1	88	0	GEO2D
	0	0	0	3	1	96	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	1	1	98	SPA3D
15%	51	45	0	0	0	4	PA
	0	82	0	18	0	0	COPY
	0	0	7	2	91	0	GEO2D
	0	0	0	2	6	92	SPA2D
	0	0	2	0	98	0	GEO3D
	0	0	0	1	5	94	SPA3D
20%	24	76	0	0	0	0	PA
	0	69	0	31	0	0	COPY
	0	0	8	2	90	0	GEO2D
	0	0	0	2	12	86	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	1	10	89	SPA3D
25%	2	98	0	0	0	0	PA
	0	53	0	46	1	0	COPY
	0	0	6	2	92	0	GEO2D
	0	0	1	4	18	77	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	1	17	82	SPA3D

Table 8. Graph Feature Vector with 100 Boosting Iterations

Another interesting observation is that the overall classification accuracy does not necessarily increase with the number of boosting iterations. It is the case that increasing the number of boosting iterations improves the classification accuracy on the unchanged data but this is not necessarily the case for the changed data. For most of the test data sets the difference is not significant but when 25% of the edges are changed, the classification accuracy is about 3% better when only 50 boosting iterations are performed as compared to 200 boosting iterations.

Another purpose to testing the robustness of the classifier is to attempt to simulate the behaviour of the classifier on unknown data. One conclusion we have, is that even if a little bit of noise is introduced into the data, the 2D models are more likely to get classified as a 3D model. The conclusion is that if unknown data is classified as a 3D model, it is possible that the correct model should be the 2D model. We also can conclude that using the graph feature vector may be more reliable than using the full feature vector.

3.2. **Classification of the Facebook networks.** Classification algorithms are built under the assumption that the test data actually belongs to one of the classes the classifier is trained to distinguish. This assumption is often not met in realistic applications, as is the case here though it is common practice to evaluate unknown data using a classification algorithm. With this in mind, it is important for us to exercise caution when interpreting our results from the Facebook data. To interpret our results, we consider how each feature contributes to the score for each model. Specifically, we are interested in the features which appear in the first layer of nodes (of depth 1) in the ADT as they are the most influential in separating the data. Furthermore, we consider how often each feature is visited when the Facebook data is put through the ADT. We also compare how well the models can generate the features exhibited by the Facebook data. In this section we give a general discussion of our results; a precise analysis of the performance of the classifier on each of the data sets can be found in Appendix A.

Our first observation is that the most significant feature in separating the classes is the assortativity coefficient. In fact, the first node in every ADT built using the full feature vector corresponds to the assortativity coefficient. This is because the assortativity coefficient of the GEO models is significantly higher than all the other models. This is due to the fact that the nodes in a GEO model have degrees which are binomially distributed, so many vertices have similar degrees. It is interesting to note that the assortativity coefficient is not included in the graph feature vector, while we just showed that the graph feature vector is equally successful in separating the models. Thus the information conveyed by the assortativity coefficient should be implicitly contained in the graphlet counts.

An important graphlet feature is g_6, which corresponds to the 4-cycle. The 4-cycle feature tends to be the most important feature overall. That is, it tends to appear frequently in the first layer of nodes in the ADT and it is usually the feature which is most visited by the Facebook data when it is put through the classifier. In some cases, the outcome of the experiment can be deduced by only considering the feature g_6.

An important difference between the models is that the PA and COPY models tend not to generate highly connected subgraphs whereas the GEO models do tend to generate highly connected subgraphs. Conversely, the PA and COPY models generate many sparse subgraphs whereas the GEO models do not. By highly connected subgraph we mean those that contain a triangle, namely: g_2, g_5, g_7. Sparse subgraphs are those without a triangle: g_1, g_3, g_4. The SPA models tend to generate a mixture of dense subgraphs and sparse subgraphs.

Classifier	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
full Princeton	-0.303	-14.551	4.599	11.287	-5.451	4.42
full American	-0.414	-12.164	-0.183	8.307	-5.578	10.025
full MIT	2.956	-12.512	2.715	13.528	-8.561	1.873
full Brown	4.998	-15.163	-0.305	1.733	-6.161	14.897
graphs Princeton	6.699	-2.227	-3.914	3.085	-3.676	0.033
graphs American	0.779	-10.639	0.381	5.834	-7.693	11.332
graphs MIT	4.097	-9.49	3.061	5.304	-2.91	-0.063
graphs Brown	6.283	-0.085	-3.774	1.827	-3.771	-0.479
other Princeton	-0.858	-3.622	-7.447	8.022	-5.029	8.941
other American	-4.612	-2.442	-3.627	6.517	-3.348	7.512
other MIT	1.956	-7.305	-2.458	2.518	-2.901	8.192
other Brown	-0.197	-3.58	-2.61	4.549	-1.606	3.44

Table 9. Scores for each Experiment for each of the classifiers with 100 boosting iterations

The overall conclusion from the performance of the classifiers on the Facebook data is that models which incorporate some preferential attachment mechanism perform best. The scores for each classifier for 100 boosting iterations can be found in Table 9. When the full feature vector is used, both SPA models appear in the top 3 in classification scores. For the Princeton and MIT experiments, SPA2D finishes first and for the American and Brown experiments, SPA3D finishes first. In all 4 experiments with the full feature vector the GEO3D and COPY models have the lowest scores. For the graph feature vector, a model with the preferential attachment mechanism always finishes first. The MIT network is the only data set where a non PA-based model (GEO2D) is in the top 3 highest scores. The ranking of the models when the other feature vector is used is consistent across all the experiments with the two SPA models are ranked 1st and 2nd and PA ranked 3rd.

Perhaps the most interesting observation comes from comparing the experiments for the Princeton and Brown networks. Though the Princeton network has 6596 vertices and the Brown network has 8600 vertices, they both have almost the same edge density. The conclusions of the two experiments are similar and for the graph feature vector in particular they are almost identical. Another observation is that the ADT's produced for each of the networks are very similar. They have the exact same first layer of nodes for both the full and graph feature vector. This suggests that training sets with graphs of the same density generate similar ADT's, and that the same classifier could be used for observed networks of comparable density. This also implies that, if the appropriate normalization factor could be found for comparing subgraph counts for the graphs of different size but similar density then the building of the classifier would only have to be done once. The same classifier could then be applied to suitably normalized feature vectors of the data. Since the generation of the samples of each model, and the computation of the graphlet counts for these samples takes a lot of computation time, this would improve our method significantly.

4. Conclusions and Further Work

The main goal of this work was to determine which of our 6 models is the most appropriate for a social network such as Facebook. The results of our experiments showed that a model incorporating a preferential attachment mechanism had the best performance. However, based on our work, it is difficult to determine whether the PA or the SPA model is better. This is because we have not performed enough experiments to develop a statistically significant sample size. However, the fact that in all four experiment, for almost every classifier generated, the PA and SPA models generally received positive scores indicate that the models do fit the test data to some degree. On the other hand, the COPY model generally gave high negative scores for almost all the classifiers generated, indicating that the model is a poor fit for the Facebook graphs considered.

Our work has shown conclusively that our classification procedure works well at separating graphs produced by each of our models even when the models generate graphs with similar degree distributions and average path lengths. This gives evidence to our claim that local structure is important in developing models for real world networks. Furthermore, we saw that the classification accuracy using all of the features and using only the graphlet features were not significantly different. We can conclude from this that considering graphlets is sufficient to separating to models.

Our results, discussed in detail in Appendix A, show that graphlets corresponding to paths, cycles and highly connected subgraphs are the most influential in distinguishing between different models. This is not a surprising conclusion because a high count of paths and a low count of complete subgraphs are characteristic of sparse models such as PA and COPY while a low count of paths and a high count of complete graphs is characteristic of denser models such as GEO2D and GEO3D. The ability of the SPA models to generate a higher number of 4-cycles then the other models results in C_4 being a distinguishing feature. The fact that the Facebook graphs tended to generate a high number of C_4's resulted in the SPA model obtaining the best score in many of the experiments.

Currently, it is necessary to generate a new training set and classifier for each test network of a given size and density. This is because graphlet counts are highly dependent on the size and density of the graph. We are interested in determining a method to normalize the graphlet counts so that graphlet counts for graphs of varying sizes and densities can be compared. Such a normalization would make it possible to build a single classifier which can test networks for a range of sizes and densities. In this work, the amount of time to perform one experiment took about two weeks so the existence of a universal classifier through the normalization of graphlet counts would make testing various real world networks a more tractable.

References

[1] J. Leskovec, C. Faloutsos, *Sampling from Large Graphs*, ACM SIGFDD International Conference on Knowledge Discovery and Data Mining, 2006.

[2] J. Leskovec, J. Kleinberg, C. Faloutsos, *Graphs over time: Densification laws, shrinking diameters and possible explanations*, ACM SIGKDD, 2005.

[3] A. Barabási, R. Albert, *Emergence of Scaling in Random Networks*, Science **286**, 1999, 509-512.

[4] J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkin, *The Web as a Graph: Measurements, Models and Methods*, Proceedings of the International Conference on Combinatorics and Computing, 1999.

[5] R.Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal, *Stochastic models for the web graph*, Proceedings of the 41th IEEE Symposium on Foundations of Computer Science, 2000.

[6] P. Krapivsky, S. Redner, *Network Growth by Copying*, Physical Review E, Vol. 71, 2005.

[7] W. Aiello, A.Bonato, C.Cooper, J. Janssen, P. Prałat, *A Spatial Web Graph Model with Local Influence Regions*, Proceedings of the 5th Workshop on Algorithms and Models for the Web-Graph, 2007.

[8] J.Janssen, *Spatial models for virtual networks*, Computability in Europe, LNCS, 2010.

[9] M. Penrose, *Random Geometric Graphs*, Oxford Studies in Prob., Vol. 5, Oxford University Press, 2003

[10] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng and B. Zhai, *Measurement-calibrated Graph Models for Social Network Experiments*, In WWW'10, 861-870, 2010.

[11] M. Middendorf, E. Ziv, C. Wiggins, *Inferring Network Mechanism: The Drosophila Melanogaster Protein Interaction Network* PNAS **102**, 2005, 3192-3197.

[12] N. Pržulj, *Biological Network Comparison using Graphlet Degree Distribution*, Bioinformatics 23, 2007, 177-183.

[13] A. Mislove, M. Marcon, K.Gummadi, P. Druschel, B. Bhattacharjee, *Measurement and Analysis of Online Social Networks*, In Proc. 7th ACM SIGCOMM Conference on Internet Measurement, 2007, 29-42.

[14] http://projects.skewed.de/graph-tool/ Home web page for the graph-tool python module.

[15] www.insna.org, Web page for the International Network for Social Network Analysis. Accessed Feb 10th, 2011.

[16] N. Shervashidze, SVN. Vishwanathan, T. Petri, K. Mehlhorn, K. Borgwardt, *Efficient Graphlet Kernels for Large Graph Comparison*, AISTATS 2009.

[17] R. Kondor, N.Shervashidze, K. Borgwardt, *The Graphlet Spectrum*, ICML, 2009.

[18] M. Newman, *Mixing Patterns in Networks*, Phys Rev. **67**, 2003.

[19] S. Wernicke, *Efficient Detection of Network graphlets*, IEEE/ACM Transaction on Computational Biology and Bioinformatics Vol 3, Issue 4, 2006, 347-359.

[20] Y. Freund, L. Mason, *The Alternating Decision Tree Learning Algorithm*, In Proc. 6th Int. Conf. on Machine Learning, 1999, 124-133.

[21] Y. Freunde, R. Schapire, *A Decision- Theoretic Generalization of Online Learning and an Application to Boosting*, 1995.

[22] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, M. Hall, *Multiclass Alternating Decision Trees*, ECML, 2002, 161-172.

[23] J. Friedman, T. Hastie, R. Tibshirani, *Additive logistic regression: a statistical view of boosting*, Annals of Statistics, 2000, 337-407.

[24] Y. Fruende, R. Schapire, *Experiments with a new boosting algorithm*, Machine Learning: Proceedings of the 13th International Conference, 148-156.

[25] http://www.cs.waikato.ac.nz/mi/weka/, Home page for the open source machine learning software Weka.

[26] C. Bishop, *Pattern Recognition and Machine Learning*, Oxford University Press, 1995.

[27] A. Traud, P. Mucha, M. Porter, *Social Structure of Facebook Networks*, arXiv:1102.2166, 2011.

Appendix A: Discussions for each Facebook Experiment

In this appendix, we discuss the experiments for each of the four social network data sets in detail.

4.1. **Princeton.** The Princeton network has 6596 vertices and 293329 edges. To interpret our results we consider the information in Tables ,10, 11, 12, 13 as well as the box plots in Figure A1. From the scores in Table 13, for the full feature vector the SPA models have the best performance. This result can be understood to a large extent by considering the feature g_6. Table 11 indicates that this is the most frequently visited feature. This feature also occurs frequently in the first layer of nodes so that it is a descriptive feature in separating the models. Comparing the number of 4-cycles in the Princeton network against range of 4-cycle counts for each of the models in Table 3, we can see that the count for the Princeton network only falls in the range of SPA2D and is very close to falling into the SPA3D range. This is the main reason that SPA2D receives the highest score. In fact, if the classification was done only from observing which models come closest to generating the number of 4-cycles in the Princeton network the ranking would be SPA2D, SPA3D, GEO2D, PA, GEO3D and COPY which is precisely the ranking that the classifier gives on the full feature vector.

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl	g1	g2	g3	g4	g5	g6	g7	g8
50 full	1	1	0	0	0	0	1	1	3	0	2	1	2	4	9	1	3
100 full	1	1	0	0	0	0	1	1	3	1	3	1	2	5	12	3	3
200 full	3	1	0	0	0	0	2	1	3	1	6	1	5	6	21	4	5

Table 10. Features visited for full feature vector for the Princeton network

Classifier	g1	g2	g3	g4	g5	g6	g7	g8
50 graph	1	0	1	2	1	1	1	0
100 graph	2	1	1	2	1	5	1	0
200 graph	2	1	3	4	4	8	4	0

Table 11. Features visited for graph feature for Princeton network

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl
50 other	0	0	0	1	0	1	1	5	6
100 other	1	1	0	2	0	1	3	12	8
200 other	1	1	1	2	2	2	3	18	12

Table 12. Features visited for other feature for Princeton network

Since a graph feature was the most influential for the full feature vector, we might expect that the SPA models should also be ranked the highest when the graph feature vector is used. From Table 13 we see that this is not the case: the PA model receives the best score with the two SPA models coming in second and third. To understand this we need to consider which feature is most influential in determining the scores for the graph feature vector. For 100 and 200 boosting iterations, Table 11 indicates that g_6 is the most frequently visited feature. But to determine which feature is the most influential we must also consider which features appear in the first layer of nodes in the ADT. In this case, for 50, 100 and 200 boosting iterations only 3 nodes appear in the first layer of nodes and 2 of these correspond to g_4 (the 4-path). By observing the box plot in Figure 3, you can see that the g_4 count for the Princeton network corresponds almost exactly to the mean count for g_4's in the PA model. Furthermore, in the ADT, the first node corresponds to g_4, which immediately leads to a score of 3.611 for the PA model and -0.722 for the other models. This gives an advantage of 4.333 for the PA model from the first node which is close to the overall difference between PA model and SPA models in the final classification score. Additionally, we can see from Figure A1 that the GEO models do not match with the number of g_4's generated by the Princeton network, which explains their poor performance. In this case, the g_4 feature turned out to be the most influential feature in determining the classification.

Classifier	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
50 full	0.232	-12.365	2.626	8.359	-3.958	5.106
100 full	-0.303	-14.551	4.599	11.287	-5.451	4.42
200 full	1.681	-21.364	5.148	15.812	-10.426	9.152
50 graphs	6.321	-0.891	-3.471	-0.025	-3.026	1.093
100 graphs	6.699	-2.227	-3.914	3.085	-3.676	0.033
200 graphs	9.377	-2.667	-3.255	3.805	-7.904	0.644
50 other	-0.728	-0.524	-3.379	2.549	-1.396	3.48
100 other	-0.858	-3.622	-7.447	8.022	-5.029	8.941
200 other	-2.97	-4.107	-12.995	9.87	-6.503	16.708

Table 13. The scores for each model, for each classifier on the Princeton network

The performance of the classifier built on the other feature vector can be easily understood by observing from Table 12, that the assortativity and the average path length are the most important features. By observing their box-plots in Figure 3, you can see that the SPA models match the average path length the best with COPY the next best and the assortativity is matched almost equally by the SPA models and the COPY model. The PA model is also close to the assortativity coefficient of the Princeton network by noticing that it is only one of the 6 which is (almost) below the assortativity of the Princeton network. In this case, the ADT is using this feature to distinguish PA from the rest of the models so many visits to assort in the tree would give a negative score to PA and a positive score for the other models. This explains why the score of PA is decreasing as the number of boosting iterations increase. The results for the other feature vector is consistent amongst all the experiments so I will not explain the results in detail for the other experiments.

4.2. **American University.** The American network has 6386 vertices and 217661 edges. The conclusions for the American University remain consistent with the conclusion from the Princeton experiment though there are some differences. It remains the case that PA and the SPA models have the best performance but the ranking of the three models differ in some areas. Let's discuss why this occurs.

Classifier	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
50 full	-0.702	-11.768	1.091	8.505	-4.532	7.398
100 full	-0.414	-12.164	-0.183	8.307	-5.578	10.025
200 full	-0.728	-18.426	-1.191	9.955	-7.246	17.625
50 graphs	0.172	-7.908	2.271	3.269	-5.6	7.792
100 graphs	0.779	-10.639	0.381	5.834	-7.693	11.332
200 graphs	4.516	-16.02	-4.241	7.002	-8.925	17.656
50 other	0.172	-7.908	2.271	3.269	-5.6	7.792
100 other	-4.612	-2.442	-3.627	6.517	-3.348	7.512
200 other	-7.279	-2.071	-5.395	8.249	-5.439	11.936

Table 14. The scores for each model, for each classifier on the American University network

The main difference that occurs at the Full feature vector in the American experiment is that SPA3D is the best and SPA2D is second. As well, the GEO model slips from 3rd place to 4th at 200 boosting iterations. Part of the explanation for why SPA3D performs better at this level is that the 4-cycle (g_6) count of the American network falls within the range of SPA3D as well as SPA2D but in Princeton it only fell within the range of SPA2D. In fact, in the Princeton network, the 4-cycle feature contributed a score of -5.3 to SPA3D while it contributes a score of 1.895 for SPA3D in the American network. This difference almost completely accounts for the discrepancies in the scores between the two experiments. The explanation for why GEO2D slips to 4th place at 200 boosting iterations can be understood by considering the feature g_7. The feature g_7 corresponds to the complete graph minus an edge and is matched well by GEO2D. This feature is visited 4 times in the Princeton classifier at 200 boosting iterations and is not visited at all in the American classifier at 200 boosting iterations. This should account for the slip in score that puts it in 4th place.

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl	g1	g2	g3	g4	g5	g6	g7	g8
50 full	0	0	1	0	0	1	0	1	2	0	1	1	4	5	8	0	4
100 full	0	0	1	1	0	1	0	1	3	0	1	1	5	7	10	0	4
200 full	0	0	1	1	0	1	0	1	3	2	5	1	8	9	16	0	4

Table 15. Features visited for full feature vector for the American network

Classifier	g1	g2	g3	g4	g5	g6	g7	g8
50 graph	1	0	0	3	4	5	1	4
100 graph	1	1	2	5	5	8	1	4
200 graph	3	1	3	8	6	12	2	5

Table 16. Features visited for graph feature for American network

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl
50 other	1	0	2	1	0	0	1	8	9
100 other	1	0	3	1	0	0	1	12	12
200 other	1	0	3	2	0	2	2	17	13

Table 17. Features visited for other feature for American network

The ranking of the models when only the graph feature vector is used is different. Instead of the ranking being PA, SPA2D, SPA3D as it was in the Princeton network, it is SPA3D, SPA2D, PA. Part of this difference can be explained because the American network visits more decision nodes when it is put through the classifier than the Princeton network visited when it was classified. In fact, the American network visits 40 nodes and the Princeton network visits 26 nodes at the 200 boosting iteration level (see Tables 11, 16). Of these nodes, 26 correspond to either g_6 or the dense subgraphs described above while 14 correspond to sparse subgraphs. Recall that the SPA models appear to be best overall at matching all the graph features; in particular they match the dense features much better than PA and COPY and they match the sparse features much better than the GEO models. It appears that for this reason and the fact that more features are visited than in the Princeton network, the influence of the g_4 feature which resulted in the top ranking of PA in the Princeton network is weakened. It is also important to note that there are 5 nodes in the first level of nodes in the ADT at 200 boosting iterations. Of these nodes, 2 correspond to g_4 and 2 correspond to g_6 with the remaining node corresponding to g_3. In the Princeton network, 2 out of 3 of the nodes in the first layer corresponded to g_4 and none of the nodes corresponded to g_6. Therefore, for this experiment, the g_6 feature is much more influential and g_4 less influential which also explains why the SPA models are ranked above PA.

4.3. **MIT.** The MIT network has 6440 vertices and 251252 edges. The results in this experiment are not completely consistent with the other experiments. In particular, the GEO2D model finishes second for both the full feature vector and the graph feature vector. It still the case that COPY and GEO3D occupy the last two positions in the rankings. We can understand the results of the experiment by studying the ADT's and see which features are influential in determining the scores. Let's first look at the full feature vector.

Classifier	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
50 full	1.23	-7.541	3.917	9.319	-6.482	-0.441
100 full	2.956	-12.512	2.715	13.528	-8.561	1.873
200 full	2.32	-13.795	2.737	17.881	-11.679	2.534
50 graphs	5.53	-5.266	-1.368	3.719	-1.17	-1.441
100 graphs	4.097	-9.49	3.061	5.304	-2.91	-0.063
200 graphs	2.905	-12.759	3.962	10.004	-2.512	-1.603
50 other	2.53	-5.236	-2.99	1.933	-2.732	6.497
100 other	1.956	-7.305	-2.458	2.518	-2.901	8.192
200 other	1.614	-8.182	-4.085	8.28	-8.158	10.537

Table 18. The scores for each model, for each classifier on the MIT network

The most significant difference is the influence of g_6. In the first layer of nodes for the tree at 200 boosting iterations there are 9 nodes and only one node corresponds to g_6. Furthermore, by observing the box plot in Figure 5, you see that the number of g_6's in the MIT network does not fall into the range of any of the models, though it comes closest to SPA2D. This is abnormal as in all other experiments the number of g_6's falls into the range of at least the SPA2D model. In Table 19 we see that the g_6 feature is still the most visited. The good performance of GEO2D could be due to the fact that not many of the sparse subgraphs are visited. Of the 31 graph-based features visited in the full feature vector, only 6 correspond to sparse subgraphs.

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl	g1	g2	g3	g4	g5	g6	g7	g8
50 full	0	1	0	0	0	0	1	1	2	0	0	1	1	1	6	3	2
100 full	1	1	0	0	0	0	1	1	2	1	2	1	1	2	9	4	3
200 full	1	1	0	0	0	0	2	2	2	1	3	1	2	4	12	5	3

Table 19. Features visited for full feature vector for the MIT network

Classifier	g1	g2	g3	g4	g5	g6	g7	g8
50 graph	0	1	0	1	2	5	0	1
100 graph	1	4	0	1	2	7	0	2
200 graph	1	4	0	2	2	9	0	5

Table 20. Features visited for graph feature for MIT network

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl
50 other	2	0	0	0	0	2	2	9	8
100 other	2	0	0	0	1	2	3	12	12
200 other	3	0	1	1	2	4	5	19	14

Table 21. Features visited for other feature for MIT network

To understand the rankings for the graph feature vector we only have to consider g_4 and g_6. At 50 boosting iterations, the PA model is the best model but at 200 boosting iterations it is ranked 3rd. This can be explained because, as explained for the Princeton network, the first node in the ADT corresponds to g_4 which gives an immediate advantage of 4.333 points for the PA model. You can see by observing Table 20, that the g_4 feature is not visited very often. Also, not many of the sparse graphs are visited so that the influence of sparse graphs greatly diminishes as more boosting iterations and therefore more decision nodes are added to the tree. The g_6 feature appears to be the most influential in determining the scores. Of the five nodes in the first layer of the ADT at 200 boosting iterations, three correspond to g_6, while the other two correspond to g_4 and g_2. The GEO2D model performs so well because of the emphasis on the denser subgraphs in determining the score. Of the 23 nodes visited by the American network at 200 boosting iterations only 3 correspond to the sparse subgraphs. Of these nodes, 9 of them correspond to g_2 and g_8 which represent K_3 and K_4. You can see that GEO2D matches these features while by observing the box plots in Figure 5. For these features, you can see that SPA2D matches them just as well as GEO2D. The American network for these features do fall into the range for the 3D versions of these network but the fit is not as good as for the 2D models. This explains why the 2D models are performing better.

4.4. **Brown.** The Brown network has 8600 vertices and 384525 edges. The results for the Brown network are similar to the results we have already seen. Interestingly, the results for the graph feature are almost the same as the results for the Princeton network. This is interesting because both networks have almost the same density though the Brown network has around 33% more vertices. We first discuss the results when the full feature vector is used.

As we have been seeing in many of the experiments, models with a preferential attachment mechanism are ranked in the top 3. The reasoning is as explained before. The PA model finishes second because by observing Table 23, you can see that the sparse graph features are visited in higher proportions than was typical in the other experiments. In particular, the g_4 feature which greatly favours PA is visited 6 times.

The most interesting result occurs for the graph feature vector. For this feature vector, the ranking of the models is identical (except for an insignificant swap of the GEO models in the final position), to what they were in the Princeton network. This is interesting because both the Princeton and the Brown networks have the same edge density. The explanation for the Princeton results holds here as well. In fact, if we consider the ADT's in both experiments for both the full and graph feature vectors, the first layer of nodes are exactly the same. This is promising to see because it indicates that training sets with the same edge density might result in the same ADT's which would greatly cut down on the amount of computational time. All that is needed is to determine an appropriate normalization for the graph features for two graphs with a different number of vertices.

Classifier	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D
50 full	3.044	-8.622	1.0	0.245	-3.191	7.522
100 full	4.998	-15.163	-0.305	1.733	-6.161	14.897
200 full	7.085	-24.459	-1.228	4.841	-9.41	23.171
50 graphs	5.114	-1.087	-2.336	1.052	-2.354	-0.388
100 graphs	6.283	-0.085	-3.774	1.827	-3.771	-0.479
200 graphs	8.672	-0.772	-6.304	3.668	-6.296	1.033
50 other	0.302	-2.357	-2.063	2.335	-0.522	2.302
100 other	-0.197	-3.58	-2.61	4.549	-1.606	3.44
200 other	1.858	-2.865	-10.064	10.21	-5.457	6.313

Table 22. The scores for each model, for each classifier on the Brown University network

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl	g1	g2	g3	g4	g5	g6	g7	g8
50 full	0	0	0	0	0	0	1	1	2	1	0	0	3	1	6	1	4
100 full	0	0	0	0	0	0	2	3	2	1	3	0	4	3	11	2	5
200 full	0	1	0	0	0	0	4	6	4	2	4	0	6	6	17	5	7

Table 23. Features visited for full feature vector for the Brown network

Classifier	g1	g2	g3	g4	g5	g6	g7	g8
50 graph	0	0	0	2	1	1	0	0
100 graph	0	0	2	2	1	1	0	0
200 graph	0	0	3	4	1	3	0	0

Table 24. Features visited for graph feature for Brown network

Classifier	d1	d2	d3	d4	d5	d6	d7	assort	apl
50 other	0	1	0	0	1	2	2	11	10
100 other	0	1	0	0	2	2	3	15	11
200 other	2	2	0	2	3	4	4	23	14

Table 25. Features visited for other feature for Brown network

Appendix B: Tables and Figures

The tables shown in this appendix give more results for the performance of the various classifiers on synthetic graphs with a certain percentage of changed edges. The results are very similar to those discussed in Section 3.1. We also include the Figures from Appendix A showing the box-plots of the features in each of our experiments.

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	90	0	10	0	GEO2D
	0	1	0	93	0	6	SPA2D
	0	0	9	0	91	0	GEO3D
	0	0	0	6	0	94	SPA3D
5%	88	8	0	0	0	4	PA
	0	100	0	0	0	0	COPY
	0	0	5	0	95	0	GEO2D
	0	0	0	57	0	43	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	3	1	96	SPA3D
10%	78	14	0	0	0	8	PA
	0	99	0	1	0	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	15	2	83	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	7	1	92	SPA3D
15%	50	47	0	0	0	3	PA
	0	92	0	8	0	0	COPY
	0	0	3	0	97	0	GEO2D
	0	0	0	8	3	89	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	6	0	94	SPA3D
20%	24	76	0	0	0	0	PA
	0	82	0	18	0	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	8	1	91	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	8	0	92	SPA3D
25%	2	98	0	0	0	0	PA
	0	67	0	33	0	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	10	0	90	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	15	0	85	SPA3D

Table 26. Full Feature Vector with 50 Boosting Iterations

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	85	1	13	1	GEO2D
	0	1	0	94	0	5	SPA2D
	0	0	2	0	98	0	GEO3D
	0	0	0	8	0	92	SPA3D
5%	88	2	0	0	0	10	PA
	0	100	0	0	0	0	COPY
	0	0	42	2	56	0	GEO2D
	0	0	1	47	0	52	SPA2D
	0	0	1	0	98	1	GEO3D
	0	0	0	3	0	97	SPA3D
10%	78	14	0	0	0	8	PA
	0	99	0	1	0	0	COPY
	0	0	5	1	94	0	GEO2D
	0	0	0	7	3	90	SPA2D
	0	0	0	0	99	1	GEO3D
	0	0	0	2	2	96	SPA3D
15%	51	45	0	0	0	4	PA
	0	92	0	8	0	0	COPY
	0	0	2	2	96	0	GEO2D
	0	0	0	6	3	91	SPA2D
	0	0	0	0	100	0	GEO3D
	0	0	0	7	7	86	SPA3D
20%	24	76	0	0	0	0	PA
	0	82	0	18	0	0	COPY
	0	0	2	2	96	0	GEO2D
	0	0	0	7	14	79	SPA2D
	0	0	0	0	100	0	GEO3D
	0	0	0	12	10	78	SPA3D
25%	2	98	0	0	0	0	PA
	0	67	0	32	1	0	COPY
	0	0	0	2	98	0	GEO2D
	0	0	0	6	20	74	SPA2D
	0	0	0	0	100	0	GEO3D
	0	0	0	15	14	71	SPA3D

Table 27. Graph Feature Vector with 50 Boosting Iterations

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	94	0	6	0	GEO2D
	0	0	0	97	1	2	SPA2D
	0	0	8	0	92	0	GEO3D
	0	0	0	0	0	100	SPA3D
5%	97	3	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	13	0	87	0	GEO2D
	0	0	0	40	0	60	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	1	0	99	SPA3D
10%	86	12	0	0	0	2	PA
	0	96	0	3	0	1	COPY
	0	0	8	0	92	0	GEO2D
	0	0	0	5	0	95	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	1	0	99	SPA3D
15%	47	50	0	0	0	3	PA
	0	84	0	14	2	0	COPY
	0	0	5	0	95	0	GEO2D
	0	0	0	6	0	94	SPA2D
	0	0	3	0	97	0	GEO3D
	0	0	0	1	0	99	SPA3D
20%	18	82	0	0	0	0	PA
	0	66	0	23	11	0	COPY
	0	0	6	0	94	0	GEO2D
	0	0	0	3	0	97	SPA2D
	0	0	6	0	94	0	GEO3D
	0	0	0	2	1	97	SPA3D
25%	1	98	0	0	0	1	PA
	0	50	13	14	23	0	COPY
	0	0	7	0	93	0	GEO2D
	0	0	0	1	1	98	SPA2D
	0	0	8	0	92	0	GEO3D
	0	0	0	1	3	96	SPA3D

TABLE 28. Full Feature Vector with 200 Boosting Iterations

Change	PA	COPY	GEO2D	SPA2D	GEO3D	SPA3D	Models
0%	100	0	0	0	0	0	PA
	0	100	0	0	0	0	COPY
	0	0	93	0	7	0	GEO2D
	0	1	0	99	0	0	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	0	0	0	100	SPA3D
5%	88	2	0	0	0	10	PA
	0	100	0	0	0	0	COPY
	0	0	45	0	55	0	GEO2D
	0	0	0	50	0	50	SPA2D
	0	0	4	0	96	0	GEO3D
	0	0	0	1	0	99	SPA3D
10%	78	14	0	0	0	8	PA
	0	97	0	3	0	0	COPY
	0	0	10	0	90	0	GEO2D
	0	0	0	4	3	93	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	0	2	0	98	SPA3D
15%	51	45	0	0	0	4	PA
	0	83	0	16	1	0	COPY
	0	0	4	0	96	0	GEO2D
	0	0	0	1	3	96	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	0	1	4	95	SPA3D
20%	24	76	0	0	0	0	PA
	0	72	0	28	0	0	COPY
	0	0	5	0	95	0	GEO2D
	0	0	0	1	10	89	SPA2D
	0	0	5	0	95	0	GEO3D
	0	0	1	0	7	92	SPA3D
25%	2	98	0	0	0	0	PA
	0	53	18	29	0	0	COPY
	0	0	6	0	94	0	GEO2D
	0	0	0	1	16	83	SPA2D
	0	0	7	0	93	0	GEO3D
	0	0	1	1	9	89	SPA3D

TABLE 29. Graph Feature Vector with 200 Boosting Iterations

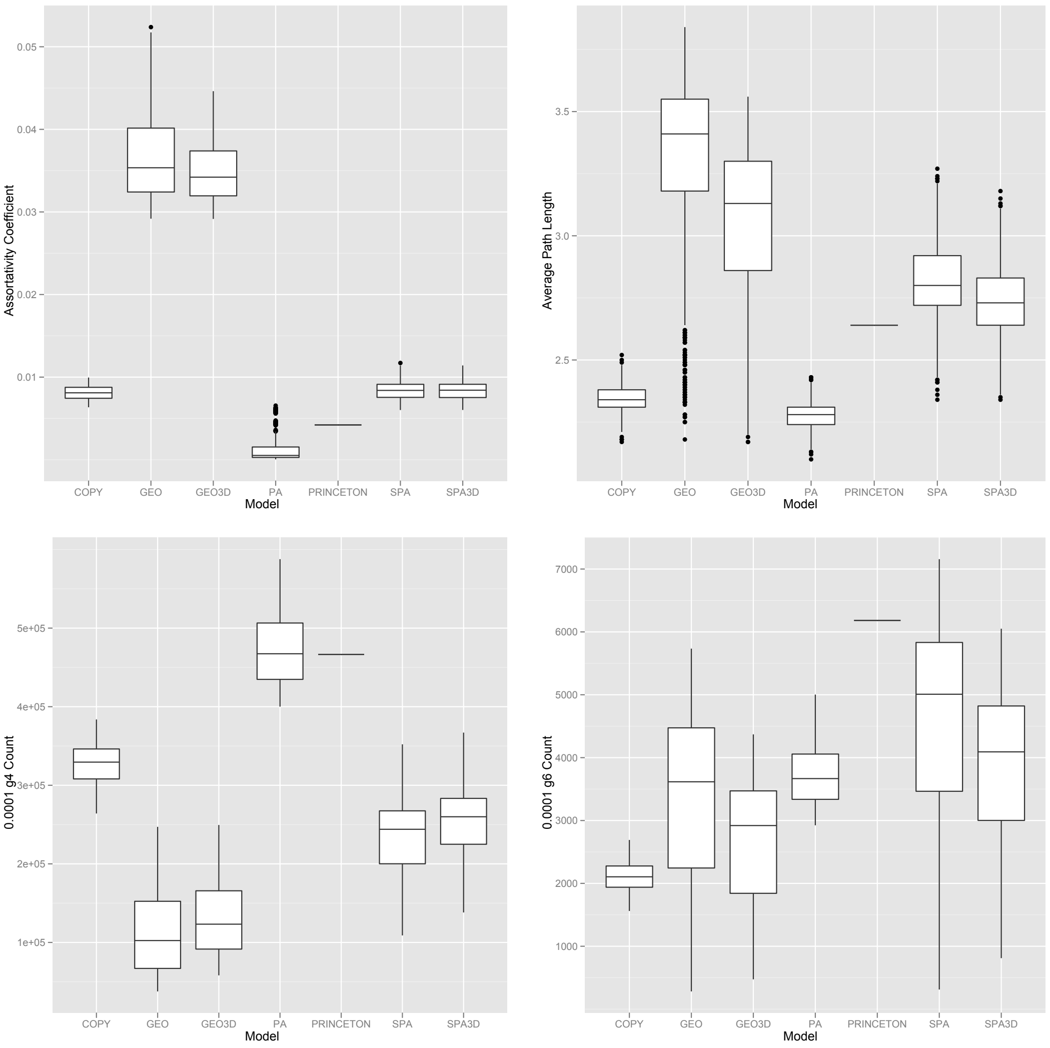

FIGURE 3. Box-plots representing the spread of the features for Princeton network.

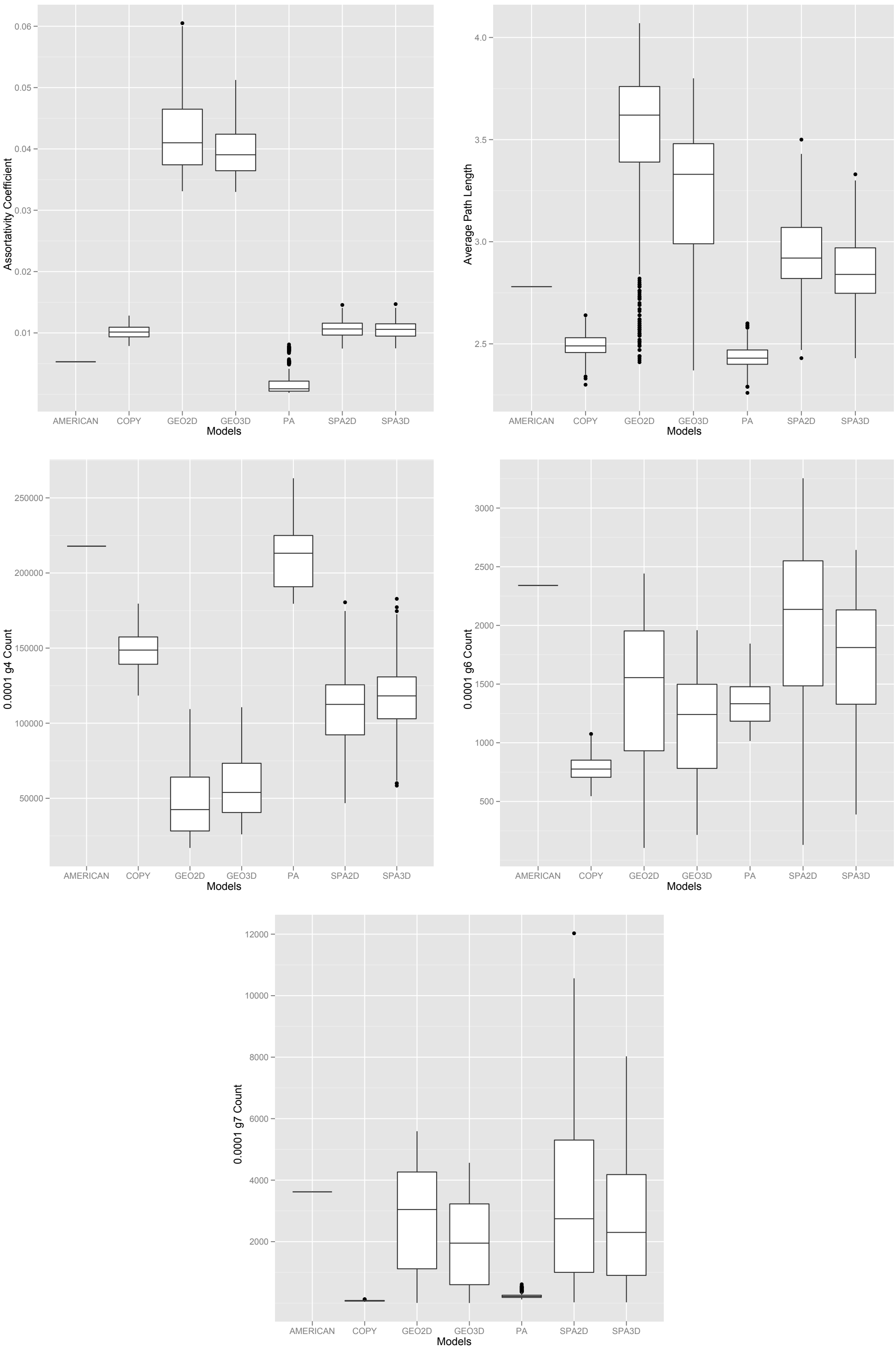

FIGURE 4. Box-plots representing the spread of the features for American network.

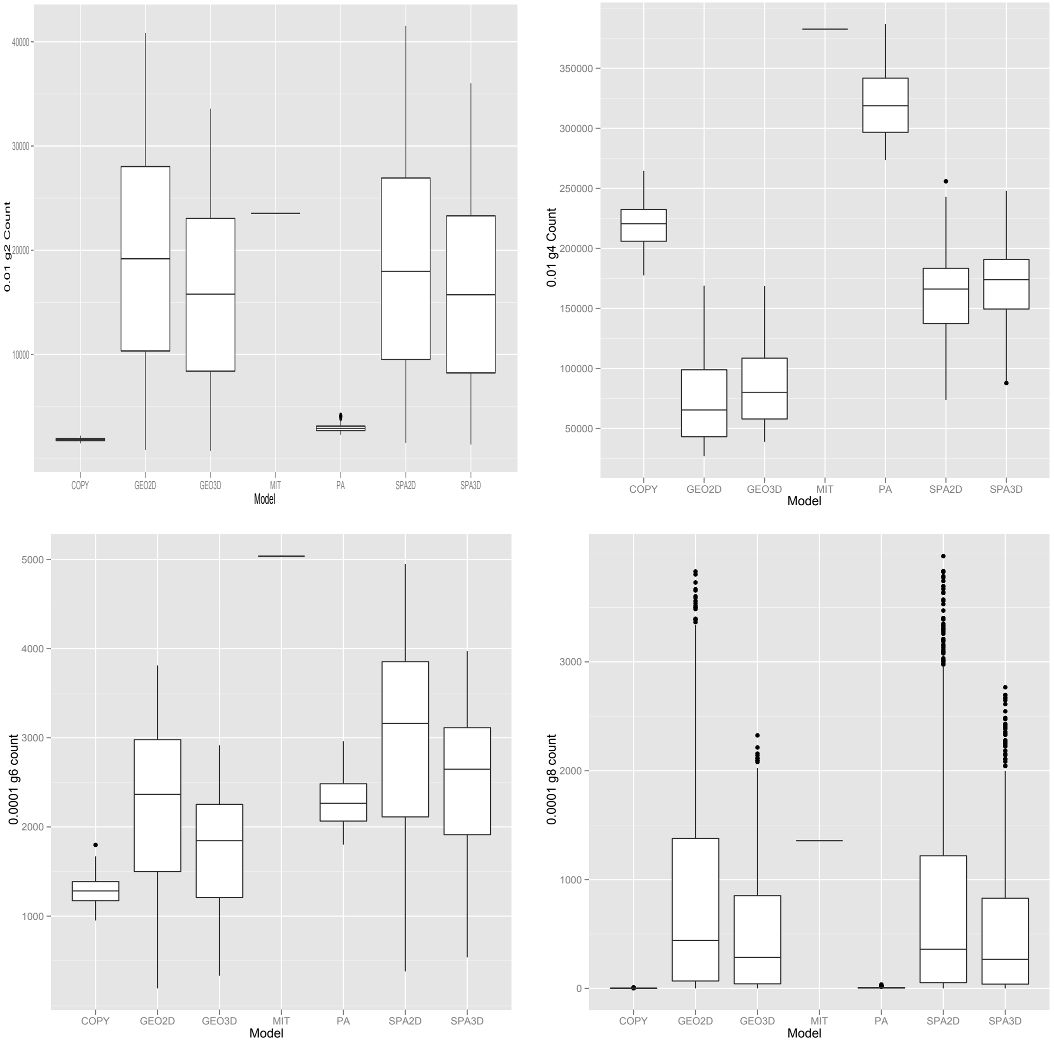

Figure 5. Box-plots representing the spread of the features for MIT network.

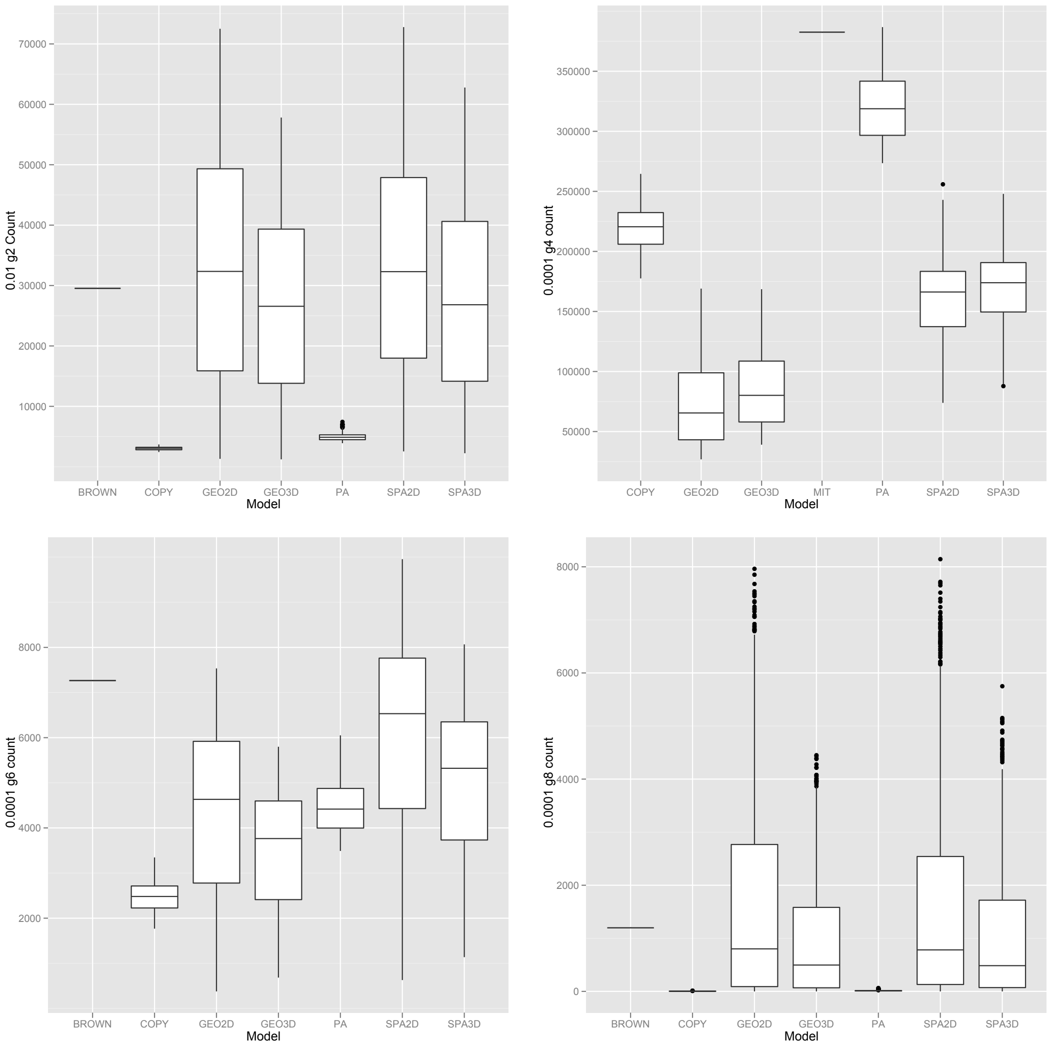

Figure 6. Box-plots representing the spread of the features for Brown network.