Network Hypothesis Testing Using Mixed Kronecker Product Graph Models

Sebastian Moreno
Computer Science Department
Purdue University
West Lafayette, Indiana 47906
Email: smorenoa@cs.purdue.edu

Jennifer Neville
Computer Science and Statistics Departments
Purdue University
West Lafayette, Indiana 47906
Email: neville@cs.purdue.edu

***Abstract*—The recent interest in networks—social, physical, communication, information, etc.—has fueled a great deal of research on the analysis and modeling of graphs. However, many of the analyses have focused on a single large network (e.g., a subnetwork sampled from Facebook). Although several studies have compared networks from different domains or samples, they largely focus on empirical exploration of network similarities rather than explicit tests of hypotheses. This is in part due to a lack of statistical methods to determine whether two large networks are likely to have been drawn from the same underlying graph distribution. Research on *across-network* hypothesis testing methods has been limited by (i) difficulties associated with obtaining a *set* of networks to reason about the underlying graph distribution, and (ii) limitations of current statistical models of graphs that make it difficult to represent variations across networks. In this paper, we exploit the recent development of *mixed-Kronecker Product Graph Models*, which accurately capture the natural variation in real world graphs, to develop a model-based approach for hypothesis testing in networks.**

***Index Terms*—Network science, graph models, hypothesis testing.**

I. Introduction

The recent interest in networks—social, physical, communication, information, etc.—has fueled a great deal of research on the analysis and modeling of graphs. However, much of this work has focused on analyzing the structure of a single large network drawn from a specific domain (e.g., Facebook). Although some of the work has compared the structure of networks from various domains (e.g., comparing biological networks to social networks), and these efforts have empirically explored the similarities (or differences) among their network statistics, there has little effort to assess whether the observed differences are statistically *significant*.

An algorithm that can determine whether two observed networks are significantly different can be useful in several contexts. For example, it can be used to identify and detect anomalous structure in networks (e.g., email, Web, Internet). It can also be used to distinguish between classes of networks that are thought to be similar but are functionally different (e.g., articulated social networks vs. interaction networks). Finally, it can be used to compare and evaluate the difference among generative graph models.

The lack of statistical hypothesis testing methods to compare across networks is due primarily to two issues. First, it is often difficult to obtain a *set* of networks to reason about the underlying distribution of a network population. Although network analysis is often conducted on *sampled* networks, time and resource constraints can make it challenging to identify and collect more than a few sample networks from a particular domain. As such, there has been little investigation of the natural variability in the characteristics of large networks sampled from a graph population. One exception is the work of [1], which reports the variation across several sets of networks that are conceivably drawn from the same population.

Second, while there has been work on statistical models that represent probability distributions over graph structures (e.g., Exponential Random Graph Models [2], Kronecker Product Graph Models [3]), there are few statistical methods to compare two networks and determine whether they are likely to have been drawn from the same underlying graph population. The work on statistical hypothesis testing for networks has primarily focused on assessing the significance of patterns *within* a network [4], [5], [6] (e.g., reciprocity) rather than comparing *across* networks.

If we can accurately estimate a probability distribution over graph space, from an observed network (or set of networks), then we can use the estimated distribution to test multiple hypotheses about the underlying system. Ideally, any probabilistic model of graphs could be used for this type of model-based approach, but many of the current models have difficulty capturing the natural variability of network populations [1] and this will result in increased Type I error (i.e., the hypothesis that two networks are drawn from the same population will be *rejected* even when it is true).

In this work, we exploit recent work on *mixed* Kronecker Product Graph Models (mKPGMs; [7], [8])—which accurately capture the structural characteristics of real world networks and their natural variation—to develop a principled approach for hypothesis testing across networks. Specifically, we develop an algorithm to test the hypothesis that two graphs are drawn from the same distribution by learning a statistical model for $P(G)$ and accepting/rejecting the hypothesis based on likelihood. We illustrate the performance of our algorithm on synthetic and real-world networks, showing that our method is able to recognize networks from the same distribution and distinguish networks from different distributions.

II. Hypothesis Testing Framework

In this work, we outline a model-based statistical hypothesis testing framework—to test the hypothesis that a new network G_{new} is likely to have been generated from the same underlying distribution as a previously observed network G_{obs}. If we can accurately estimate a probability distribution over graph space, i.e., $P(G)$, from an observed network (or set of networks), then we can use the estimated distribution to test multiple hypotheses about the data and underlying system. Thus, robust and flexible statistical models of graph populations can form the foundation for principled methods of hypothesis testing in network domains.

General hypothesis testing procedures work as follows. Formulate a null hypothesis and alternative hypothesis. Identify a statistic T that can be used to assess the truth of the null hypothesis. Determine the distribution of T under the null hypothesis. Calculate the value of the statistic from the observed data, i.e., t_{obs}. Compute the probability p that a value more extreme than t_{obs} would be observed under the null hypothesis. If $p < \alpha$, the null hypothesis is rejected.

To test hypotheses across networks, we will model the underlying "normal" distribution from which G_{obs} was sampled and use the likelihood of the network as the test statistic T. Specifically, we will use a statistical model M to estimate $P(G)$ from G_{obs}. Based on an accurate estimate of $P(G)$, we can then test the null hypothesis that G_{obs} and G_{new} are drawn from the same distribution by simply determining whether G_{new} was likely to have been generated from the estimated $P(G)$. If the likelihood is low, we reject the null and determine that G_{new} was sampled from a different distribution.

There are a few issues, however, that complicate the implementation of this basic hypothesis testing approach for networks. First, since graph space is discrete, there is no natural ordering over likelihoods that can be used to determine the probability of observing more extreme values than a particular statistic t_{obs} (i.e., $P(G_{new})$). In addition, for some graph models, it is difficult to calculate the likelihood of an observed network efficiently. We will address the first issue by generating networks from $P(G)$ to construct an empirical sampling distribution of likelihoods. The second issue will be addressed through the development of alternative likelihood functions (see Section III).

More specifically, our framework assumes as input a statistical model M, which can be used to accurately estimate $P(G)$ and efficiently sample from the estimated distribution. Given an input network G_{obs}, the method uses M to learn $\tilde{P}(G)$ (ideally capturing the natural variation in the overall population). The algorithm then generates a set of graphs $\mathbf{G}$ from $\tilde{P}(G)$, calculates their likelihoods, and sorts them to form an empirical sampling distribution. The resulting distribution will be used to test hypotheses by determining a threshold on the value of $\tilde{P}(G)$ to identify extreme values of likelihood as follows. Let N_g the number of samples in the set of graphs $\mathbf{G}$ (i.e., $|\mathbf{G}| = N_g$). Then we choose a network G_k from $\mathbf{G}$ such that $|\{G_i : \tilde{P}(G_i) < \tilde{P}(G_k)\}| = \alpha \cdot N_g$ and set the threshold as $t = \tilde{P}(G_k)$. Figure 1 shows a graphical illustration of this approach for two graphs G_{new} and G_{obs}. The null hypothesis is rejected for G_{new} due to its low likelihood but the null is accepted for G_{obs} due to its higher likelihood.

Algorithm 1 Hypothesis testing algorithm

```
Require: $G_{obs}$, $N_g$, $G_{new}$, $\alpha$
 1: $M$ = Learn a model from $G_{obs}$
 2: for $i = 1; i++; i \leq N_g$ do
 3:     Generate network $G_i$ by sampling from $M$
 4: end for
 5: Generate empirical sampling distribution $\tilde{P}(G)$ from $G_1, \cdots, G_{N_g}$
 6: Set $t = \tilde{P}(G_k)$ s.t. $|\{G_i : \tilde{P}(G_i) < \tilde{P}(G_k)\}| = \alpha \cdot N_g$
 7: if $\tilde{P}(G_{new}) \leq t$ then
 8:     RETURN REJECT $H_0$
 9: end if
10: RETURN ACCEPT $H_0$
```

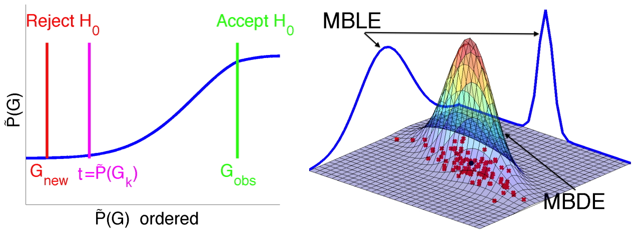

Fig. 1. Left: graphical illustration of the hypothesis testing framework. Right: generated and training network characteristics (red dots and black dot respectively) modeled by multivariate/independent Gaussian(s) for approximation of $\tilde{P}$ using MBDE and MBLE.

Algorithm 1 outlines the pseudocode for our hypothesis testing algorithm. The algorithm learns a model M from the training network G_{obs} (line 1). Then, the empirical sample distribution $\tilde{P}(G)$ and a threshold t are generated/estimated according to N_g networks sampled from M (line 2-6). The hypothesis with respect to the network G_{new} is tested using its likelihood (line 7). If $\tilde{P}(G_{new})$ is greater than t, the null hypothesis is accepted (and the algorithm concludes that G_{new} is sampled from $P(G)$), otherwise the null is rejected.

III. Implementation

Above we outlined a general algorithm for model-based hypothesis testing in networks. In this section, we discuss details of our specific implementation.

A. Generative model

In principle any statistical model of graphs that can accurately estimate $P(G)$ from data could be used as the input model M. However, since much of the recent work on developing probabilistic graphs models has focused on evaluating whether the mode of the estimated distribution is close to a single target network, the graphs generated from these models typically exhibit significantly less variance than what is observed in real data [1]. This is a critical limitation for developing robust statistical tests, because if the variance of $P(G)$ is *underestimated* it will lead to high *Type I* error (i.e., the null hypothesis will often be rejected even when the test network is generated from the same underlying graph distribution).

To avoid a high Type I error rate, in our implementation we use a statistical model capable of learning both the salient properties of the training network and the variation in the underlying population—the *mixed Kronecker Product Graph Model* (mKPGM; [7], [8]). mKPGMs use parameter tying to accurately capture the clustering and natural variation observed in real-world networks. Given parameters Θ, K, and $\ell \in [1, K]$, where K is the number of Kronecker multiplications, ℓ is the number of untied levels, and Θ is a $b \times b$ parameter matrix ($\forall i,j\ \Theta(i,j) \in [0,1]$), the mKPGM algorithm generates a network of size b^K as follows. The algorithm makes ℓ Kronecker multiplications of Θ with itself to compute a probability matrix P_ℓ, from which it generates an adjacency matrix A_ℓ with each $A_\ell(i,j)$ sampled independently from a $Bernoulli(P_\ell(i,j))$. Then, the algorithm uses a subsequent Kronecker product, $A_\ell \otimes \Theta$, to obtain a new probability matrix $P_{\ell+1}$. To tie parameters, an adjacency matrix $A_{\ell+1}$ is sampled from $P_{\ell+1}$ before calculating any further Kronecker products. This process is repeated $K - \ell - 1$ times to generate the final adjacency matrix A_K (of 0s and 1s) that represents the final network. For details on how to learn mKPGMs see [8].

Beside its ability to capture variation in $P(G)$, another strength of mKPGM is its sampling method which can generate a network in $O(E)$ time. We exploit this characteristic to generate graphs from the learned model and efficiently estimate an empirical sampling distribution over graphs $\hat{\mathcal{G}}$.

B. Score function

Unfortunately mKPGMs calculate the likelihood of an observed network by averaging over all its possible permutations. This is intractable to compute exactly in large-scale networks. Moreover, sampling permutations does not produce accurate enough approximations of the likelihood to support hypothesis testing. To address this issue, we develop an alternative likelihood score based on the observed properties of the networks in the empirical sampling distribution.

Given a specified set of N_m network characteristics ($X_1, \cdots, X_{N_m}$), we propose a score function which estimates an alternative distribution $\tilde{P}$ based on N_m. We consider approximating $\tilde{P}$ in two different ways:

Model based density estimation (MBDE): The empirical distribution of the graph statistics is approximated by a multivariate normal distribution with parameters μ and Σ (Figure 1). The parameters are estimated by the average and covariance of the moments over the N_g generated networks in $\hat{\mathcal{G}}$. With $\hat{\mu} = \overline{\mathbf{X}}$ and $\hat{\Sigma} = Q_{N_g}$, the likelihood of G_{new} is calculated as: $\tilde{P}(G_{new}) = P\left(X_1 < x_1^o, \cdots, X_{N_m} < x_{N_m}^o \,|\, \mathcal{N}(\overline{\mathbf{X}}, Q_N)\right)$, where x_i^o is the value of the i^{th} moment of G_{new}.

Model based likelihood estimation (MBLE): The empirical distribution of each graph statistic is independently approximated with a normal distribution $\mathcal{N}_i \sim (\mu_i, \sigma_i)$ (Figure 1). With $\hat{\mu}_i = \overline{X}_i$ and $\hat{\sigma}_i = S_i$, the likelihood of G_{new} is calculated as: $\tilde{P}(G_{new}) = \prod_i^{N_m} P\left(X_i < x_i^o \,|\, \mathcal{N}_i(\overline{X}_i, S_i)\right)$, where x_i^o is the value of the i^{th} moment of G_{new}.

Since N_m moments are computed for each G_i, the selected moments are crucial for the algorithm. A large number of moments will increase the type I error due to the curse of dimensionality. In this work, we consider three characteristics: X_1: avg. degree, X_2: avg. geodesic distance, and X_3: global clustering coefficient. We select these characteristics because they are common choices for discriminating among social networks and they can be efficiently estimated ($O(E)$ if we sample to estimate path lengths). Moreover, across networks they all can be modeled by Gaussians (normality tests *Jarque-Bera* and *Lilliefors* were confirmed in most cases).

We note that if multiple network samples are available for training, then the alternative score functions could be estimated directly from the data without the use of a model M. As such, we compare to this baseline in our experiments. However, our method assumes that we have only one observed network to learn the model—recall that since the full network is often not publicly available or is costly to acquire, it can be difficult for researchers to collect even a single network sample.

C. Algorithm

The implementation of Algorithm 1, using mKPGM and score functions MBDE or MBLE, requires two slight modifications to the pseudocode. The first modification occurs after line 3—once a network G_i is generated, its set of moments $\mathbf{X}^{T_i} = (X_1, \cdots, X_{N_m})$ are calculated. The second modification occurs in line 5—$\tilde{P}(G)$ is estimated from $\mathbf{X}^{T_1}, \cdots, \mathbf{X}^{T_{N_g}}$ using MBDE or MBLE instead of the likelihood directly.

IV. Methodology

To investigate the proposed framework's utility for hypothesis testing, we consider scenarios where the underlying distributions are known and quantitatively evaluate the classifications made by the method's accept/reject decisions.

A. Datasets

We evaluate our approach on three types of datasets—synthetic networks, email networks, and social networks.

In the first experiment, we use synthetic data to confirm that our algorithm can discriminate different types of network data. We generate four different sets of data using mKPGM model with $K = 11$ (2^{11} nodes), $\ell = 8$, and the Θs listed in Figure 2. Θ_1 and Θ_2 are selected for training purposes, generating a single training network (9,382 and 10,364 edges for Θ_1 and Θ_2 respectively). We also generate 100 test networks from each setting to use for evaluation.

We show two of the characteristics of the synthetic data in Figure 2. Note that the data generated from Θ_1 and Θ_4 overlap, which it makes difficult to determine if networks from these two spaces are sampled from the same distribution. The data generated from Θ_2 and Θ_3 have similar characteristic as well.

In the second experiment, we consider a dataset constructed from anonymized Purdue email logs, consisting of email transactions among Purdue users observed over 189 days (08/22/11 to 02/28/12). In order to remove the effects of mailing lists and automated emails, we dropped any node with zero incoming or

Θ_1	$\begin{bmatrix} 0.98 & 0.58 \\ 0.58 & 0.18 \end{bmatrix}$
Θ_2	$\begin{bmatrix} 0.18 & 0.98 \\ 0.98 & 0.18 \end{bmatrix}$
Θ_3	$\begin{bmatrix} 0.58 & 0.58 \\ 0.58 & 0.58 \end{bmatrix}$
Θ_4	$\begin{bmatrix} 0.99 & 0.63 \\ 0.63 & 0.05 \end{bmatrix}$

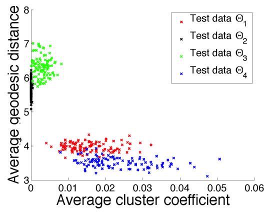

Fig. 2. Synthetic data parameters and characteristics.

outgoing degree. The final dataset consists of 68,339 different users and 3,755,129 email links. From this, we constructed a set of 189 *daily* network snapshots for our samples. In this experiment, for training we used a random *weekday* network (February 16th, 2012; 11,813 nodes and 27,676 edges).

The third experiment considers two sets of static networks from different underlying graph distributions—the public Purdue Facebook network (Facebook) and social networks from the National Longitudinal Study of Adolescent Health (AddHealth). The Facebook data consists of a set of over 50,000 Purdue Facebook users with over 400,000 wall links among those users, over a year-long period. From this data, we sampled 50 temporal networks of size 2,187 based on the same process described in [7]. The AddHealth dataset consists of a set of social networks from 25 different schools in the National Longitudinal Study of Adolescent Health survey (see [9] for major details). In this work, we considered the schools with network sizes varying from 800 to 2,000 nodes. For training, we picked a single network randomly from each dataset.

B. Baseline and comparison models

We compared our methods against five baseline models:

Degree Distribution (DD)**:** The null hypothesis that the observed cumulative degree distribution (CDF_2) is the same as a previously observed CDF_1, is based on two-sample Kolmogorov–Smirnov test [10] and is rejected if $KS(CDF_1, CDF_2) > c(\alpha)\sqrt{\frac{N_1+N_2}{N_1N_2}}$. Here $\alpha = 3\%$, N_i is the number of elements to estimate CDF_i, $c(\alpha)$ is determined by [10], and $KS(CDF_1, CDF_2) = max_x(|CDF_1(x) - CDF_2(x)|)$.

Percentage Network Characteristics (PNC)**:** Two networks are considered dissimilar if the distance between them (defined as the average over the percentage difference of each characteristic) differs more than 5%.

Network Characteristics Mahalanobis distance (NC)**:** The null hypothesis that the observed network was generated from the same underlying graph distribution as the training network is rejected if $D_{MH} > \chi^2_{(p,1-\alpha)}$, where $\alpha = 3\%$, and D_{MH} is the Mahalanobis distance between network characteristics.

Data Empirical Distribution (DED): The null hypothesis that two networks are sampled from the same distribution is tested using our basic hypothesis testing framework, but k-nearest density estimation is used to model $P(G)$.

Data Multivariate Gaussian (DMG): The null hypothesis that two networks are sampled from the same distribution is tested using our basic hypothesis testing framework, but a multivariate normal distribution is used to model $P(G)$.

Note that three of the baseline methods (NC, DED and DMG) are not useful when only a single network is available for training due to lack of (co)variance. For this reason, we vary the number of training networks available to these methods in the experiments to determine the number needed to achieve comparable result to our model-based approach, which requires only one training network.

In addition, since our proposed framework can use any probabilistic model of graphs that can learn $P(G)$ and generate networks for the empirical sampling distribution, we compare our choice of mKPGM against three alternative graph models:

Kronecker Product Graph Model (KPGM) [3]. The KPGM is a special case of the mKPGM with $\ell = K$.

Chung Lu Model (CL) [11]. The CL algorithm models the expected degree distribution via a set of weights w_i for each node. The edges of a network are sampled independently with $Bernoulli(w_i w_j)$.

Exponential Random Graph Model (ERGM) [2]. ERGMs represent graph distributions with an exponential linear model that uses feature counts. We use alternating k-stars and k-paths.

C. Evaluation

To evaluate performance we assign a class label to every network based on which underlying distribution it was drawn from. Then for each approach on each dataset, we calculate the *Type I* error rate and *statistical power* (i.e., 1-*Type II* error). The *Type I* error is the percentage of cases when the test network (G_{new}) is sampled from the underlying distribution but the hypothesis test is rejected. *Type I* error varies between 0 and 1, with a value of α indicating that the method correctly reflects the chosen critical threshold. Statistical power is the percentage of cases when the test network is not sampled from the underlying null distribution and the hypothesis test is correctly rejected. This value also varies between 0 and 1, with a value of near 1 indicating that the method correctly distinguishes networks generated from different distributions. If two methods both have Type I errors near α, the method that maximize statistical power is preferable.

V. Experimental Results

This section reports the results of the three experiments described above. For synthetic data, we report the Type I error and statistical power. However, for real data we report the false positive rate (% of networks from the alternative distribution that were classified as coming from the null) and the specificity (% of networks from the alternative distribution that were correctly classified). These differences in reporting are due to our knowledge of the data. For synthetic data, we know the true underlying graph distribution $P(G)$ from which the data is generated. However, for real networks, we do not know the true $P(G)$, so we assign class labels based on best assessment of information in the data (weekdays and weekend labels for Email experiment, and Facebook and AddHealth labels for Facebook-AddHealth experiment).

Model	Type I error			Statistical power		
Train	–	Θ_1	Θ_2	–	Θ_1	Θ_2
Tested	Θ_1	Θ_1	Θ_2	Θ_4	Θ_4	Θ_3
$mKPGM_{MBDE}$	3%	**13%**	**8%**	86%	94%	100%
$mKPGM_{MBLE}$	3%	32%	**8%**	81%	96%	99%
DD	–	22%	44%	–	44%	100%
PNC	–	99%	8%	–	97%	62%

TABLE I
SYNTHETIC RESULTS

A. Synthetic experiment

Table I shows the synthetic experiment results. The 2nd and 5th columns show the results when we use the true parameters of the mKPGM (rather than learning them) and test on networks generated from Θ_1 and Θ_4. The empirical *Type I* error correctly matches the expected level since $\alpha = 3\%$ (see III-B). The observed statistical power (86% and 81%) are also expected due to network similarity in Θ_1 and Θ_4 (see Fig. 2).

The 3rd and 6th columns show the results when an mKPGM is trained on a network from Θ_1 and the test considers a network from Θ_1 and Θ_4. The $mKPGM_{MBDE}$ and $mKPGM_{MBLE}$ exhibit low *Type I* errors, but in each case the error is higher than α. This effect is due to some error in the mKPGM learning process, which may not recover the true parameters exactly. However, the low *Type I* error and the high statistical power confirm that $mKPGMs$ are able to learn the underlying graph distribution of Θ_1. In contrast, DD and PNC do not achieve the same results. Even though DD has low *Type I* error, its low statistical power means it is unable to differentiate between these types of networks. In contrast, PNC exhibits high *Type I* error, thus it does not model the underlying distribution well.

The 4th and 7th columns show the results when an mKPGM is trained on a network from Θ_2 and the test considers a network from Θ_2 and Θ_3. Again the $mKPGMs$ exhibit low *Type I* errors and high statistical power, confirming their ability to accurately model the underlying graph distribution. Here DD exhibits high Type I error and while PNC has low Type I error, it also has low statistical power.

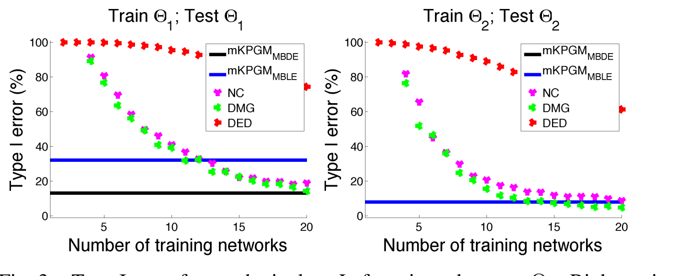

Fig. 3. Type I error for synthetic data. Left: train and test on Θ_1. Right: train and test on Θ_2. mKPGMs solid lines represent a single training network.

Figure 3 shows that baseline models, NC, DMG and DED, require at least 11 training networks to achieve the same level of Type I error as $mKPGM_{MBDE}$.

B. Email experiment

Considering that most professional email interaction among academics occurs during weekdays, we test our approach under the assumption that *weekday* networks are generated from a different distribution than *weekend* networks.

Model	False positive rate	Specificity
Train	Weekdays	Weekdays
Test	Weekdays	Weekends
$mKPGM_{MBDE}$	**12%**	77%
$mKPGM_{MBLE}$	14%	77%
DD	68%	94%
PNC	89%	100%
$KPGM_{MBDE}$	100%	100%
$KPGM_{MBLE}$	100%	100%
$ERGM_{MBDE}$	100%	100%
$ERGM_{MBLE}$	99%	100%
CL_{MBDE}	100%	100%
CL_{MBLE}	100%	100%

TABLE II
EMAIL RESULTS.

Table II reports the results on email data. Besides the DD and PNC, we also show our proposed framework using different graph models ($mKPGM$, $KPGM$, CL, and $ERGM$). $mKPGM_{MBDE}$ and $mKPGM_{MBLE}$ obtain a low false positive rate and a high specificity. This confirms that mKPGMs learn the underlying graph distribution and recognize most weekday networks as being generated from the underlying "weekday" distribution, as well as correctly distinguishing most weekend networks. $KPGM$, CL, and $ERGM$ show a high false positive rate, indicating that these models do not learn the underlying graph distribution accurately. Baselines models DED and PNC also show a high false positive rate, which results in most weekday networks being classified as weekends.

We examine the rejected weekday and accepted weekend networks for $mKPGMs$. Most rejected weekday networks are related to academic breaks, deadlines or holidays, when the normal traffic of emails is reduced/changed (e.g., Fall/Winter Break, Thanksgiving). In contrast, the accepted weekend networks are related to important deadlines for professors and students (e.g., the last/first weekends of fall/spring semesters).

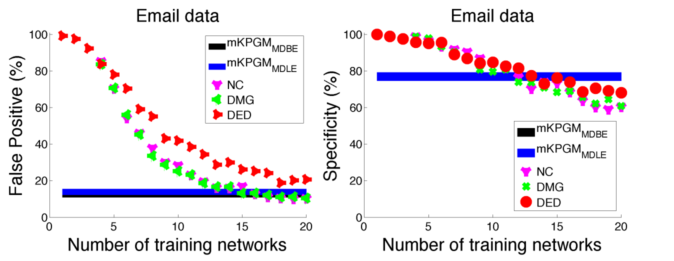

Fig. 4. Percentage of false positives rates (left) and Specificity (right) for Email data. mKPGMs solid lines represent a single training network.

The left plot of Figure 4 shows that baseline models NC, DMG and DED need at least 15 networks to achieve the same false positive rate as our model-based approach with $mKPGMs$. However, with 15 training networks their specificity is less than that of $mKPGMs$ (right plot Fig. 4).

C. Facebook-AddHealth experiment

Here we evaluate if the approach can differentiate between the Facebook and AddHealth social network datasets. We learn the model on a single network from each dataset (Facebook network 46—2,187 nodes and 5,632 edges; AddHealth network 7—1,051 nodes and 6,354 edges), and then test on the remaining networks in both datasets.

Model	False positive rate		Specificity	
Train	Facebook	AddHealth	Facebook	AddHealth
Test	Facebook	AddHealth	AddHealth	Facebook
$mKPGM_{MBDE}$	8%	32%	100%	100%
$mKPGM_{MBLE}$	8%	20%	100%	100%
DD	16%	84%	100%	100%
PNC	**0%**	**4%**	100%	100%
$KPGM_{MBDE}$	100%	100%	100%	100%
$KPGM_{MBLE}$	100%	100%	100%	100%
$ERGM_{MBDE}$	100%	100%	100%	100%
$ERGM_{MBLE}$	96%	100%	100%	100%
CL_{MBDE}	100%	100%	100%	100%
CL_{MBLE}	100%	100%	100%	100%

TABLE III
FACEBOOK–ADDHEALTH RESULTS.

The results are reported in table III. Again we see that $mKPGMs$ can discriminate among networks drawn from different graph distributions. $mKPGMs$ achieve, in most cases, the smallest false positive rate and highest specificity. The high false positive rate for AddHealth is due to the low clustering coefficient of the training network. Similar to previous results, when we use KPGM, CL, and/or ERGM the null hypothesis is rejected for almost all networks. The DD baseline exhibits good performance when trained on Facebook, but poor performance on AddHealth. PNC obtains a low false positive rate, while keeping a high specificity. This can be explained due to the dissimilarities between Facebook and AddHealth networks rather than correct learning of the underlying distribution.

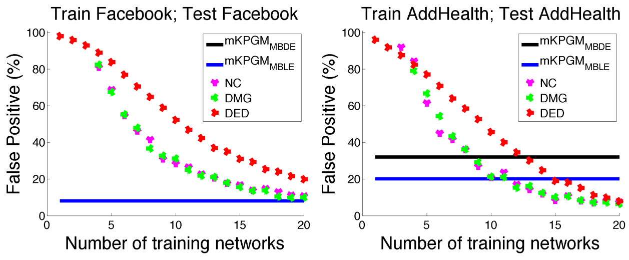

Fig. 5. Percentage of false positives rates for Facebook (left) and AddHealth (right) data. mKPGMs solid lines represent a single training network.

Figure 5 shows that baselines models NC, DMG and DED require more than 36% of the network samples to achieve a false positive rate comparable to $mKPGMs$, which are trained with a single network.

In summary, $mKPGMs$ are able to learn the underlying graph distribution from a single training network in both synthetic and real datasets. Our two methods ($mKPGM_{MBDE}$ and $mKPGM_{MBLE}$) outperform, in almost all cases, all of the baselines methods. The lack of (co)variance for NC, DED, and DMG, when they are trained on a small number of networks, leads to high *Type I* error. Moreover, in contrast to alternative graph models (KPGMs, CLs, and ERGMs), mKPGMs learn the *variance* of the underlying distributions correctly, and thus achieve reasonable *Type I* error levels.

VI. CONCLUSIONS

We presented a novel hypothesis testing framework to determine if a new network is sampled from the same underlying graph distribution as a previously observed network. Our algorithm, based on mKPGMs, is able to learn a model of the structure of a network *population*, which is then used to generate an empirical sampling distribution of the underlying graph distribution and identify similar networks.

We tested our proposed framework on three types of networks. The results confirm that our algorithm, implemented with mKPGMs, learns the underlying graph distribution along with its respective variance, from a single network observation. This enables the correct classification of networks that were drawn from the same network population, while networks generated from alternative distributions result in the correct rejection of the null hypothesis. In contrast, most of our baseline models obtained worse results than $mKPGMs$, even though we utilized the same network characteristics for testing. Moreover, the NC, DMG and DED baselines required a large number of observed networks to accurately capture the variation of the underlying graph distribution. Finally, we investigated implementations of our framework based on $KPGM$, CL, and $ERGM$ and unfortunately the results show the models cannot learn the underlying distribution $P(G)$ accurately enough to use as a basis for hypothesis testing.

In conclusion, our $mKPGM_{MBDE}$ and $mKPGM_{MBLE}$ methods are accurate, efficient, and robust—making them useful for analysis in network domains where only a few networks are available for learning a model of "normal" behavior. Our proposed approach is a first step towards the development of a more principled hypothesis testing framework for networks.

ACKNOWLEDGMENT

This research is supported by NSF and DARPA under contract number(s) IIS-1219015, IIS-1017898, IIS-1149789, and N660001-1-2-4014. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF, DARPA or the U.S. Government.

REFERENCES

[1] S. Moreno and J. Neville, "An investigation of the distributional characteristics of generative graph models," in *Proceedings of WIN'09*, 2009.
[2] S. Wasserman and P. E. Pattison, "Logit models and logistic regression for social networks: I. An introduction to Markov graphs and p*," *Psychometrika*, vol. 61, pp. 401–425, 1996.
[3] J. Leskovec and C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07, 2007, pp. 497–504.
[4] T. A. B. Snijders, "Markov chain monte carlo estimation of exponential random graph models," *Journal of Social Structure*, vol. 3, 2002.
[5] G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison, "Recent developments in exponential random graph (p*) models for social networks," *Social Networks*, vol. 29, pp. 192–215, 2007.
[6] D. R. Hunter, S. M. Goodreau, and M. S. Handcock, "Goodness of fit for social network models," *JASA*, pp. 248–258, 2008.
[7] S. Moreno, S. Kirshner, J. Neville, and S. Vishwanathan, "Tied kronecker product graph models to capture variance in network populations," in *Allerton'10*, 2010, pp. 17–61.
[8] S. I. Moreno, J. Neville, and S. Kirshner, "Learning mixed kronecker product graph models with simulated method of moments," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '13, 2013, pp. 1052–1060.
[9] K. Harris, "The National Longitudinal Study of Adolescent health, Waves I to III, 1994-2002 [machine-readable data file and documentation]," *University of North Carolina at Chapel Hill.*, 2008.
[10] F. J. Massey, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, pp. 68–78, 1951.
[11] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *PNAS*, vol. 99, no. 25, pp. 15 879–15 882, 2002.