Internet Mathematics Vol. 8, No. 3: 232–256

Moment-Based Estimation of Stochastic Kronecker Graph Parameters

David F. Gleich and Art B. Owen

Abstract. Stochastic Kronecker graphs supply a parsimonious model for large sparse real-world graphs. They can specify the distribution of a large random graph using only three or four parameters. Those parameters have, however, proved difficult to choose in specific applications. This article looks at method-of-moments estimators that are computationally much simpler than maximum likelihood. The estimators are fast, and in our examples, they typically yield Kronecker parameters with expected feature counts closer to a given graph than we get from KronFit. The improvement is especially prominent for the number of triangles in the graph.

1. Introduction

Stochastic Kronecker graphs were introduced in [Leskovec et al. 05] as a method for simulating very large random graphs. Random synthetic graphs are used to test graph algorithms and to understand observed properties of graphs. Using simulated graphs instead of real measured ones, it is possible to test algorithms on graphs larger or denser than presently observed ones. Simulated graphs also

ISSN: 1542-7951 print

allow one to judge which features of a real graph are likely to hold in other similar graphs and which are idiosyncratic to the given data.

Stochastic Kronecker graphs are able to serve these purposes through a model that has only three or four parameters. Parameter estimation poses unique challenges for those graphs. The main problem is that for a graph with N nodes, the likelihood has contributions from $N!$ permutations of the nodes [Leskovec and Faloutsos 07]. In practice, many thousands or millions of randomly sampled permutations are used to estimate the likelihood. Even then it takes more than $O(N^2)$ work to evaluate the likelihood contribution from one of the permutations.

In this paper we present a method-of-moments strategy for parameter estimation. While moment methods can be inefficient compared to maximum likelihood, statistical efficiency is of reduced importance for enormous samples and in settings in which the dominant error is lack of fit. The method equates expected-to-observed counts for edges, triangles, hairpins (2-stars or wedges), and tripins (3-stars). The Kronecker model gives quite tractable formulas for these moments.

The outline of this paper is as follows. Section 2 defines Kronecker graphs and introduces some notation. Section 3 derives the expected feature counts. Section 4 describes how to solve method-of-moment equations for the parameters of the Kronecker graph model. Section 5 presents some examples on fitting Kronecker models to some real-world graphs. We compare several moment-based ways to estimate Kronecker graph parameters and find that the most reliable results come from a criterion that sums squared relative errors between observed and expected features. We find that the fitted Kronecker models usually underestimate the number of triangles compared to the real graphs. While our parameter estimates underestimate triangle counts and some other features, we find that they provide much closer matches than some previously published parameters fit by KronFit. Section 6 fits parameters to graphs that were randomly generated from the Kronecker model. We find that the estimated parameters closely track their generating values, with some small bias when a parameter is at the extreme range of valid values. Section 7 has our conclusions.

The data for our examples can be found online at http://www.cs.purdue.edu/homes/dgleich/codes/kgmoments along with the code used to estimate Kronecker parameters.

2. The Kronecker Model

Given a node set $\mathcal{N}$ of cardinality $N \geq 1$ and a matrix $P_{ij} \in [0,1]$ defined over $i, j \in \mathcal{N}$, a random graph $G^*(P)$ is one in which the edge $[ij]$ exists with probability P_{ij} and all N^2 edges do or do not exist independently. The graph G^* includes loops $[ii]$ and may possibly include both $[ij]$ and $[ji]$. It is a steppingstone to our desired graph G. We snip out loops and double edges from G^* to get the desired random graph $G(P)$ with edges $[ij]$ only when $i \neq j$ and $[\max(i,j), \min(i,j)] \in G^*$, using any nonrandom ordering of $\mathcal{N}$. We assume that P is a symmetric matrix, and so the ordering of nodes does not affect the distribution.

The description of P allows up to $N(N-1)/2$ parameters that affect the outcome. Much more parsimonious descriptions can be made by taking P to be the Kronecker product of two or more smaller matrices. Recall that the Kronecker product of matrices $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{r\times s}$ is

$$X \otimes Y \equiv \begin{pmatrix} X_{11}Y & X_{12}Y & \cdots & X_{1n}Y \\ X_{21}Y & X_{22}Y & \cdots & X_{2n}Y \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1}Y & X_{m2}Y & \cdots & X_{mn}Y \end{pmatrix} \in \mathbb{R}^{mr\times ns}.$$

An extremely parsimonious stochastic Kronecker graph takes P to be the r-fold Kronecker product of $\Theta = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, for $a, b, c \in [0,1]$. That is,

$$P = P^{(r)} = \Theta \otimes \Theta \otimes \cdots \otimes \Theta \equiv \Theta^{[r]}.$$

If the power r is known, then only three numbers need to be specified, and with them we can then simulate other graphs that are like the original. Perhaps surprisingly, stochastic Kronecker graphs imitate many, but of course not all, of the important features seen in large real-world graphs. See, for example, [Leskovec and Faloutsos 07].

We would like to pick parameters $a, b, c \in [0,1]$ to match the properties seen in a real and large graph. The properties we will consider are some expected feature counts, according to an objective function developed in the next two sections. There are several ways that one could define a matching criterion between observed and expected moments. Section 4.2 presents a family of such criteria. In standard statistical settings with smaller data sets it is often best to use likelihood. We ended up settling on a sum over features of squared relative errors.

Parameter matrices $\Theta = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} c & b \\ b & a \end{smallmatrix}\right)$ both give rise to the same graph distribution. To force identifiability, we may assume that $a \geq c$.

We close this section with some additional references. The giant component for Kronecker graphs was characterized in [Horn and Radcliffe 11]. A related result about the number of triangles in a Kronecker graph appears in [Tsourakakis 08]. In [Seshadhri et al. 11], the authors study and fix issues with the degree distribution of a Kronecker graph and empirically illustrate a misfit between k-core results on Kronecker graphs and real-world networks. Finally, in [Palla et al. 10], the authors derive a graph from a singular measure on $[0,1]^2$, where that measure is defined as a limit of tensor products, resembling the Kronecker model with a very large power r.

3. Moment Formulas

The Kronecker structure in P makes certain aspects of G very tractable. For example, the number E of edges in G can be shown to have expectation

$$\mathbb{E}\,(E) = \frac{1}{2}\big((a+2b+c)^r - (a+c)^r\big). \tag{3.1}$$

Simply counting the edges in G gives us valuable information on the parameter vector (a,b,c). Because E is a sum of independent Bernoulli random variables, we find that $\mathbb{V}\mathrm{ar}(E) \le \mathbb{E}\,(E)$, and so the relative uncertainty $\sqrt{\mathbb{V}\mathrm{ar}(E)}/\mathbb{E}\,(E) \le \mathbb{E}\,(E)^{-1/2}$ will be small in a graph with a large number of expected edges.

This section derives equation (3.1) and similar formulas for the expected number of features of various types. The expected feature counts require sums over various sets of nodes. Section 3.1 records some summation formulas that simplify that task. Then Section 3.2 turns expected feature counts into sums, and Section 3.3 shows how those sums simplify for stochastic Kronecker matrices.

3.1. Summation Formulas

Let $i,j,k,l \in \mathcal{N}$ for a finite index set $\mathcal{N}$. A plain summation sign $\sum$ represents sums over all combinations of levels of all the subscripting indices used. The symbol $\sum^*$ includes all levels of all indices, except for any combinations in which two or more of those indices take the same value. In several places we find that sums are easier to perform over all levels of all indices, while the desired sums are over unique levels. Here we record some formulas to translate the second type into the first.

It is elementary that

$$\sum_{ij}{}^{*} f_{ij} = \sum_{ij} f_{ij} - \sum_{i} f_{ii}, \tag{3.2}$$

and similarly

$$\sum_{ijk}{}^{*} f_{ijk} = \sum_{ijk} f_{ijk} - \sum_{ij} (f_{ijj} + f_{iji} + f_{iij}) + 2\sum_{i} f_{iii}. \tag{3.3}$$

When there are four indices, we get

$$\begin{aligned}\sum_{ijkl}^{*} f_{ijkl} &= \sum_{ijkl} f_{ijkl} - \sum_{ijk}\Big(f_{ijki} + f_{ijkj} + f_{ijkk} + f_{ijik} + f_{ijjk} + f_{iijk}\Big) \\ &\quad + \sum_{ij}\Big(2\left(f_{ijjj} + f_{ijii} + f_{iiji} + f_{iiij}\right) + f_{ijij} + f_{ijji} + f_{iijj}\Big) - 6\sum_{i} f_{iiii}.\end{aligned} \tag{3.4}$$

Equation (3.4) is more complicated than the others. It can be proved by defining $g_{ijk} = \sum_l f_{ijkl} - f_{ijki} - f_{ijkj} - f_{ijkk}$, writing $\sum_{ijkl}^{*} f_{ijkl} = \sum_{ijk}^{*} g_{ijk}$, and then applying (3.3).

In some of our formulas below, the first index is singled out, but the others are interchangeable. By this we mean that $f_{ijk} = f_{ikj}$ when there are three indices, while $f_{ijkl} = f_{ijlk} = f_{ikjl} = f_{iklj} = f_{iljk} = f_{ilkj}$ is the version for four indices.

When indices after the first are interchangeable, then equation (3.3) simplifies to

$$\sum_{ijk} f_{ijk} - \sum_{ij}\Big(f_{ijj} + 2f_{iij}\Big) + 2\sum_{i} f_{iii}, \tag{3.5}$$

and equation (3.4) simplifies to

$$\sum_{ijkl} f_{ijkl} - 3\sum_{ijk}\Big(f_{iijk} + f_{ijjk}\Big) + \sum_{ij}\Big(2f_{ijjj} + 5f_{iijj} + 4f_{iiij}\Big) - 6\sum_{i} f_{iiii}. \tag{3.6}$$

When all indices ijk are exchangeable, so that $f_{ijk} = f_{ikj} = f_{jik} = f_{jki} = f_{kij} = f_{kji}$, then equation (3.5) simplifies to

$$\sum_{ijk} f_{ijk} - 3\sum_{ij} f_{iij} + 2\sum_{i} f_{iii}. \tag{3.7}$$

3.2. Expected Feature Counts for Independent Edges

The graph features we describe are shown in Figure 1. In addition to edges, there are hairpins (2-stars) in which two edges share a common node, tripins (3-stars) in which three edges share a node, and triangles. The Kronecker model has independent edges. Here we find the expected feature counts for any random graph where edge $[ij]$ appears with probability P_{ij} and edges are independent.

Recall that G^* is a random graph with $\Pr([ij] \in G^*) = P_{ij}$ (independently). Let it have incidence matrix A^*. That is, $A^*_{[ij]} = 1$ if $[ij] \in G^*$ and is zero otherwise. There may be loops $A^*_{ii} = 1$, and for $i \neq j$, A^*_{ij} and A^*_{ji} are independently generated. The graph G is formed by deleting loops from G^* and symmetrizing

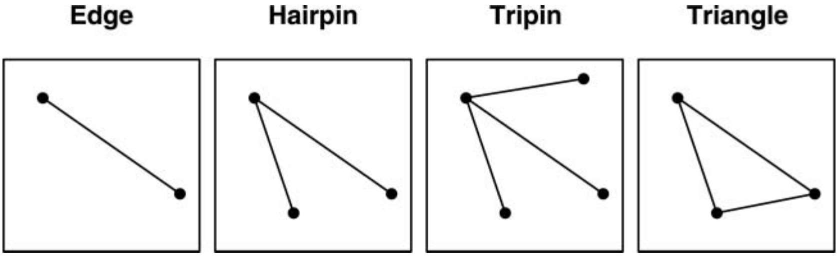


Figure 1. Illustration of some of the graph features that we can count, for use in moment-based estimates of the parameters in the stochastic Kronecker graph.

the incidence matrix via

$$A_{ij} = \begin{cases} A^*_{ij}, & i > j, \\ 0, & i = j, \\ A^*_{ji}, & i < j. \end{cases}$$

The number of edges in G is $E = \frac{1}{2}\sum^*_{ij} A_{ij}$. The expected number of edges satisfies

$$2\,\mathbb{E}\,(E) = \mathbb{E}\left(\sum\nolimits^*_{ij} A_{ij}\right) = \sum_{ij}{}^* P_{ij} = \sum_{ij} P_{ij} - \sum_{i} P_{ii}, \tag{3.8}$$

using $\mathbb{E}\,(A_{ij}) = \mathbb{E}\,(A^*_{ij})$.

The number of hairpins in G is $H = \frac{1}{2}\sum^*_{ijk} A_{ij}A_{ik}$. Dividing by two adjusts the sum for counting $\{[ij],[ik]\}$ twice. The expected value of H satisfies

$$2\,\mathbb{E}\,(H) = \sum_{ijk}{}^* P_{ij}P_{ik} = \sum_{ijk} P_{ij}P_{ik} - \sum_{ij} P_{ij}^2 - 2\sum_{ij} P_{ii}P_{ij} + 2\sum_{i} P_{ii}^2$$

by letting $f_{ijk} = P_{ij}P_{ik}$, for which $f_{ijk} = f_{ikj}$, and applying equation (3.5).

The number of triangles in G is $\Delta = \frac{1}{6}\sum^*_{ijk} A_{ij}A_{ik}A_{jk}$, because the sum counts each triangle $3! = 6$ times. The expected value of each term is $f_{ijk} = P_{ij}P_{ik}P_{jk}$, which is symmetric in its three arguments, and so we may apply equation (3.7) to get

$$6\,\mathbb{E}\,(\Delta) = \sum_{ijk} P_{ij}P_{ik}P_{jk} - 3\sum_{ij} P_{ii}P_{ij}^2 + 2\sum_{i} P_{ii}^3.$$

The number of tripins in G is $T = \frac{1}{6}\sum^{*}_{ijkl} A_{ij}A_{ik}A_{il}$. The final three indices in $f_{ijkl} = P_{ij}P_{ik}P_{il}$ are exchangeable, and so equation (3.6) applies. Thus

$$\begin{aligned}6\,\mathbb{E}\,(T) = &\sum_{ijkl} P_{ij}P_{ik}P_{il} - 3\sum_{ijk} P_{ii}P_{ij}P_{ik} - 3\sum_{ijk} P_{ij}^2 P_{ik}\\ &+ 2\sum_{ij} P_{ij}^3 + 5\sum_{ij} P_{ii}P_{ij}^2 + 4\sum_{ij} P_{ii}^2 P_{ij} - 6\sum_{i} P_{ii}^3.\end{aligned}$$

3.3. Simplifying the Sums

The sums in the expected counts simplify because of the properties of the Kronecker graph. Let the node set be $\mathcal{N} = \mathcal{N}_r = \{0, 1, \ldots, 2^r - 1\}$. For $i \in \mathcal{N}$ write $i = \sum_{s=1}^{r} 2^{s-1} i_s$ for $i_s \in \{0, 1\}$. Similarly, let j, k, and l be described in terms of $j_s, k_s, l_s \in \{0, 1\}$ for $s = 1, \ldots, r$.

The matrix entry $P_{ij} = P_{ij}^{(r)}$ may be written

$$P_{ij}^{(r)} = \prod_{s=1}^{r} \Theta_{i_s j_s}.$$

For $r \geq 2$, we simplify the expression by induction using a smaller version of the problem defined via $P^{(r-1)}$. Specifically,

$$\begin{aligned}\sum_{ijk} P_{ij}^{(r)} P_{ik}^{(r)} &= \sum_{i_1} \cdots \sum_{i_r} \sum_{j_1} \cdots \sum_{j_r} \sum_{k_1} \cdots \sum_{k_r} \prod_{s=1}^{r} \Theta_{i_s j_s} \Theta_{i_s k_s}\\ &= \Bigl(\sum_{i_1} \cdots \sum_{i_{r-1}} \sum_{j_1} \cdots \sum_{j_{r-1}} \sum_{k_1} \cdots \sum_{k_{r-1}} \prod_{s=1}^{r-1} \Theta_{i_s j_s} \Theta_{i_s k_s} \Bigr) \sum_{i_r j_r k_r} \Theta_{i_r j_r} \Theta_{i_r k_r}\\ &= \Bigl(\sum_{ijk} P_{ij}^{(r-1)} P_{ik}^{(r-1)} \Bigr) \sum_{i_r j_r k_r} \Theta_{i_r j_r} \Theta_{i_r k_r}\\ &= \Bigl(\sum_{i_r j_r k_r} \Theta_{i_r j_r} \Theta_{i_r k_r} \Bigr)^{r},\end{aligned}$$

where indices i_s, j_s, and k_s are summed over their full ranges, and the indices i, j, k for $P_{ij}^{(r-1)} P_{ik}^{(r-1)}$ are summed over the node set $\mathcal{N}_{r-1} = \{0, \ldots, 2^{r-1} - 1\}$.

All of the sums of products of elements of $P_{ij}^{(r)}$ listed in the previous section, with summation over all levels of each index, also reduce in this way to rth powers of their value for the case $r = 1$.

For $r = 1$ we need to sum products of elements of P over i or over i, j or over i, j, k. These cases correspond to the first two, four, or eight rows of Table 1 for $\Theta = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$. For instance, $\mathbb{E}\,(H)$ requires $\sum_{ijk} P_{ij}P_{ik}$, which we know to be the

i	j	k	Θ_{ii}	Θ_{ij}	Θ_{ik}	Θ_{jk}
0	0	0	a	a	a	a
1	0	0	c	b	b	a
0	1	0	a	b	a	b
1	1	0	c	c	b	b
0	0	1	a	a	b	b
1	0	1	c	b	c	b
0	1	1	a	b	b	c
1	1	1	c	c	c	c

Table 1. Entries in the matrix Θ with various indexing patterns needed in the examples. Sums over i, ij, and ijk use, respectively, the first two, four, and eight rows of the table.

rth power of

$$\begin{aligned}\sum_{i_r j_r k_r} \Theta_{i_r j_r} \Theta_{i_r k_r} &= a^2 + ba + b^2 + cb + ab + b^2 + bc + c^2 \\ &= (a+b)^2 + (b+c)^2.\end{aligned} \tag{3.9}$$

The first expression (3.9) follows by summing over the eight rows of Table 1. As a result,

$$\sum_{ijk} P_{ij} P_{ik} = \left((a+b)^2 + (b+c)^2\right)^r.$$

In the rest of this section, we record the other sums we need. First, the sums over one index variable take the form

$$\sum_i P_{ii}^m = \left(a^m + c^m\right)^r, \tag{3.10}$$

where cases $m = 1, 2, 3$ are used in our expected feature counts. The sums over two index variables are

$$\sum_{ij} P_{ii}^m P_{ij}^n = \left(a^m (a^n + b^n) + c^m (b^n + c^n)\right)^r.$$

The cases we need are for $(m, n) \in \{(0,1), (0,2), (0,3), (1,1), (1,2)\}$.

Four sums over three indices are used. They are

$$\begin{aligned}
\sum_{ijk} P_{ij}P_{ik} &= \big((a+b)^2+(b+c)^2\big)^r,\\
\sum_{ijk} P_{ij}^2P_{ik} &= \big(a^3+c^3+b(a^2+c^2)+b^2(a+c)+2b^3\big)^r,\\
\sum_{ijk} P_{ij}P_{ik}P_{jk} &= \big(a^3+c^3+3b^2(a+c)\big)^r,\\
\sum_{ijk} P_{ii}P_{ij}P_{ik} &= \big(a(a+b)^2+c(b+c)^2\big)^r.
\end{aligned}$$

Finally, one sum over four indices is used:

$$\sum_{ijkl} P_{ij}P_{ik}P_{il} = \big((a+b)^3+(b+c)^3\big)^r.$$

3.4. Expected Feature Counts

Now we can specialize the results of Section 3.2 to the Kronecker graph setting. Gathering together the previous developments, we obtain

$$\begin{aligned}
2\,\mathbb{E}(E) &= \big(a+2b+c\big)^r - \big(a+c\big)^r,\\
2\,\mathbb{E}(H) &= \big((a+b)^2+(b+c)^2\big)^r - 2\big(a(a+b)+c(c+b)\big)^r\\
&\quad - \big(a^2+2b^2+c^2\big)^r + 2\big(a^2+c^2\big)^r,\\
6\,\mathbb{E}(\Delta) &= \big(a^3+3b^2(a+c)+c^3\big)^r - 3\big(a(a^2+b^2)+c(b^2+c^2)\big)^r + 2\big(a^3+c^3\big)^r,\\
6\,\mathbb{E}(T) &= \big((a+b)^3+(b+c)^3\big)^r - 3\big(a(a+b)^2+c(b+c)^2\big)^r\\
&\quad - 3\big(a^3+c^3+b(a^2+c^2)+b^2(a+c)+2b^3\big)^r + 2\big(a^3+2b^3+c^3\big)^r\\
&\quad + 5\big(a^3+c^3+b^2(a+c)\big)^r + 4\big(a^3+c^3+b(a^2+c^2)\big)^r - 6\big(a^3+c^3\big)^r.
\end{aligned}$$

In each formula, the terms from sums over fewer indices come after those from more indices. The later terms adjust for loops and double edges and other degenerate quantities. For large r, we expect that the first term should be the most important. In particular, if $\min(a,b,c)>0$, then in all cases, the first quantity raised to the power r is the largest one. For example, the first term in $\mathbb{E}(E)$ is $(1+2b/(a+c))^r$ times as large as the second one, which subtracts out loops.

The first term will dominate for large r unless $b \ll a+c$. The relative magnitude of the second term is

$$\left(\frac{a+c}{a+2b+c}\right)^r = 2^{r\log_2((a+c)/(a+2b+c))} = N^{-\alpha},$$

where $\alpha = \log_2((a+2b+c)/(a+c))$. If $\alpha > 1/2$, then dropping the second term in $\mathbb{E}(E)$ makes a smaller difference than the sampling uncertainty in E. This holds when the off-diagonal element of Θ is not too small compared to the average of the diagonal elements: $b > (\sqrt{2}-1)(a+c)/2$.

3.5. Illustrations

Some special cases of the formulas are of interest. For example, if $b = 0$, then there are no edges in G^* apart from loops. As a result, G has 2^r isolated nodes. We find from the above that $\mathbb{E}\,(E) = \mathbb{E}\,(H) = \mathbb{E}\,(\Delta) = \mathbb{E}\,(T) = 0$ when $b = 0$.

If instead $a = c = 0$, then each node $i \in \mathcal{N}$ with coordinates $i_1, \ldots, i_r$ has a dual node i^* that has coordinates $i_s^* = 1 - i_s$ for $s = 1, \ldots, r$. The only possible edges in G are between nodes and their duals. There are $N = 2^r$ nodes each with probability b^r of having an edge out to its dual. The formula above gives $\mathbb{E}\,(E) = (2b)^r/2 = Nb^r/2$ when $a = c = 0$, as it should. We also get $\mathbb{E}\,(H) = \mathbb{E}\,(\Delta) = \mathbb{E}\,(T) = 0$ when $a = c = 0$.

If $a = b = c = 1$, then G^* has every possible edge and loop with probability 1. As a result, G is the complete graph on $N = 2^r$ nodes. Then it has $N(N-1)/2$ edges, $N(N-1)(N-2)/2$ hairpins, $N(N-1)(N-2)/2$ triangles, and it has $N(N-1)(N-2)(N-3)/6$ tripins.

4. Solving for a, b, and c

Now we turn to the problem of estimating a, b, and c from a real-world graph G though to be a random graph from a distribution similar to the Kronecker distribution. There are four equations in Section 3.4. To estimate a, b, and c will require at least three of them. Because they are high-order polynomials, it is possible that there are multiple solutions or even none at all. The latter circumstance would provide some evidence of lack of fit of the stochastic Kronecker model to a given graph. Regardless, each of the equations involves the count of a feature in the graph.

4.1. Counting Features in a Graph

Three of the features we use are easily obtainable from the degrees of the nodes. Let $d_i = \sum_{j \in \mathcal{N}} A_{ij}$ be the degree of node i in graph G. Then

$$E = \frac{1}{2}\sum_i d_i,$$

$$H = \frac{1}{2}\sum_i d_i(d_i - 1),$$

$$T = \frac{1}{6}\sum_i d_i(d_i - 1)(d_i - 2)$$

give the number of edges, hairpins (or wedges), and tripins in terms of the degrees d_i.

The number of triangles Δ is not a simple function of d_i. Algorithms to count triangles are considered in [Schank and Wagner 05] and [Tsourakakis 08]. The time complexity can be as low as $O(E^{3/2})$, and sometimes even lower for approximate counting [Kolountzakis et al. 10].

4.2. Objective Functions

A pragmatic way to choose a, b, and c is to solve

$$\min_{a,b,c} \sum_F \frac{(F - \mathbb{E}_{a,b,c}(F))^2}{\mathbb{E}_{a,b,c}(F)}, \tag{4.1}$$

where the sum is over three or four of the features $F \in \{E, H, T, \Delta\}$ from Section 3.4, and the minimization is taken over $0 \le c \le a \le 1$ and $0 \le b \le 1$. The terms in (4.1) are scaled by an approximate variance. A sharper expression would account for correlations among the features used. That should increase statistical efficiency, but in large problems, lack of fit to the Kronecker model is likely to be more important than inefficiency of estimates within it.

Many real-world networks may not have good fits in terms of these three Kronecker parameters. This is the case for most of the forthcoming experiments. The following more general objective can be more robust to these deviances:

$$\min_{a,b,c} \sum_F \frac{D(F, \mathbb{E}_{a,b,c}(F))}{N(F, \mathbb{E}_{a,b,c}(F))}. \tag{4.2}$$

Here D is either of the following two distance functions:

$$D_{\text{sq}}(x,y) = (x-y)^2 \quad \text{and} \quad D_{\text{abs}}(x,y) = |x-y|,$$

and N is one of the normalizations

$$N_F(F, \mathbb{E}) = F, \quad N_{F^2}(F, \mathbb{E}) = F^2, \quad N_{\mathbb{E}}(F, \mathbb{E}) = \mathbb{E}, \quad N_{\mathbb{E}^2}(F, \mathbb{E}) = \mathbb{E}^2.$$

Using D_{sq} and $N_{\mathbb{E}}$ makes it equal to the previous objective (4.1).

In principle, either of the two distance functions can be combined with any of the four normalizations. We do not think it reasonable to expect a quadratic denominator to be a suitable match for the absolute error. Therefore, our investigations exclude combination of D_{abs} with either N_{F^2} or $N_{\mathbb{E}^2}$.

We will find in Section 5 below that robust results arise from the combination D_{sq} and N_{F^2}, for which (4.2) reduces to

$$\min_{a,b,c} \sum_F \left(\frac{F - \mathbb{E}_{a,b,c}(F)}{F} \right)^2, \tag{4.3}$$

a sum of squared relative errors. Relative errors are more stable than absolute ones for quantities of very different magnitudes.

Because there are only three parameters, the criterion (4.2) can simply be evaluated over a grid inside $\{(a,b,c) \in [0,1]^3 \mid a \geq c\}$. To be sure of taking a point within ε of the minimizer takes work $O(\varepsilon^{-3})$. An alternative is to employ a general nonlinear minimization procedure. The remainder of this section looks at a method to reduce that effort.

4.3. Matching Leading Terms

In a synthetic graph, $N = 2^r$ is known. In fitting to a real-world graph, a pragmatic choice is $r = \lceil \log_2(N) \rceil$. The interpretation is that the random graph G^* may have had isolated nodes that were then dropped in forming G, but we suppose that fewer than half of the nodes in G^* have been dropped.

If we considered just the leading terms, then we could get estimates $\hat{a}$, $\hat{b}$, and $\hat{c}$ by solving three of the equations

$$\begin{aligned} e \equiv (2E)^{1/r} &= \hat{a} + 2\hat{b} + \hat{c}, \\ h \equiv (2H)^{1/r} &= (\hat{a} + \hat{b})^2 + (\hat{b} + \hat{c})^2, \\ \delta \equiv (6\Delta)^{1/r} &= (\hat{a}^3 + \hat{c}^3) + 3\hat{b}^2(\hat{a} + \hat{c}), \\ t \equiv (6T)^{1/r} &= (\hat{a} + \hat{b})^3 + (\hat{b} + \hat{c})^3. \end{aligned}$$

The equations for e and h together can be solved to get

$$\begin{aligned} \hat{x} \equiv \hat{a} + \hat{b} &= \frac{e + \sqrt{2h - e^2}}{2}, \\ \hat{y} \equiv \hat{b} + \hat{c} &= \frac{e - \sqrt{2h - e^2}}{2}, \end{aligned} \qquad (4.4)$$

where we have assumed that $a \geq c$. The transformed tripin count t matches $\hat{x}^3 + \hat{y}^3$, and so it is redundant given e and h if we are just using leading terms. We must either count triangles or use higher-order terms.

Equation (4.4) may fail to have a meaningful solution. At a minimum, we require $e^2 \leq 2h$ and $e \geq \sqrt{2h - e^2}$. These translate into

$$h \leq e^2 \leq 2h,$$

which is equivalent to

$$2H \leq 4E^2 \leq 2^{r+1} H,$$

which in terms of node degrees is

$$\sum_i d_i(d_i - 1) \leq \Big(\sum_i d_i\Big)^2 \leq N \sum_i d_i(d_i - 1). \qquad (4.5)$$

The left-hand inequality in (4.5) holds for every graph, but the right-hand side need not. It holds when $N^{-1}\sum_i(d_i-\bar{d})^2 \geq \bar{d} = N^{-1}\sum_i d_i$. If the variance of the node degrees d_i is smaller than their mean, then equation (4.4) does not have real-valued solutions. The degree distribution of a stochastic Kronecker graph has heavy tails [Mahdian and Xu 07]. Therefore, in applications in which that model is suitable, equation (4.4) will give a reasonable solution.

When the d_i have a variance larger than their mean, then we can do a univariate grid search for $b \in [0,1]$ using equation (4.4) to get $a = x - b \equiv a(b)$ and $c = y - b \equiv c(b)$. The choice of b can then be made as the minimizer of $\left|a(b)^3 + c(b)^3 + 3b^2(a(b)+c(b)) - \delta\right|$.

5. Examples

In this section, we experiment with different techniques for fitting the parameters of the Kronecker model. These experiments involve eight real-world networks whose statistical properties are listed in the rows of Tables 2 through 4 labeled "Source." All of the graphs have had self-loops removed. The reported edge counts represent undirected edges, that is, half the number of nonzeros in the adjacency matrix.

The networks ca-GrQc, ca-HepTh, ca-HepPh are coauthorship networks from arXiv [Leskovec et al. 07]. The nodes of the network represent authors, and there is an edge between two nodes when the authors jointly wrote a paper. These files were downloaded from http://snap.stanford.edu/data (the SNAP website). Likewise, the hollywood-2009 network is a collaboration graph between actors and actresses in IMDB [Boldi et al. 11, Boldi and Vigna 04]. It was downloaded from http://law.dsi.unimi.it/datasets.php. Nodes are actresses or actors, and edges are collaborations on a movie, as evidenced by jointly appearing in the cast roster. These networks are naturally undirected and all edges are unweighted.

Both as20000102 and as-Skitter are technological infrastructure networks [Leskovec et al. 07]. The data were again downloaded from the SNAP website. Each node represents a router in the Internet, and edges represent a physical or virtual connection between the routers. Again, these networks are undirected and unweighted.

The wikipedia-20051105 graph is a symmetrized link graph of the articles on Wikipedia generated from data download on November 5, 2005 [Constantine and Gleich 07]. The underlying network is directed, but in these experiments, we have converted it into an undirected network by dropping the direction of the edges. The graph file can be downloaded from UFL repository http://www.cise.ufl.edu/research/sparse/matrices/Gleich/.

Graph	Kroneck. Parameters			Graph/Expected Features					
Fit type	a	b	c	Vertices	Edges	Hairpins	Tripins	Triangles	Obj.
Stochastic Kronecker									
Source	0.99	0.48	0.25	16384	30830	521676	8659050	854	—
$D_{\text{sq}}, N_{\mathbb{E}}$	0.993	0.476	0.255	16384	1.00	1.00	1.000	1.0010	$7.76{\cdot}10^{-1}$
$D_{\text{sq}}, N_{\mathbb{E}^2}$	0.993	0.476	0.254	”	1.00	1.00	1.001	1.0000	$9.72{\cdot}10^{-6}$
D_{sq}, N_F	0.993	0.476	0.255	”	1.00	1.00	1.000	1.0014	$7.80{\cdot}10^{-1}$
D_{sq}, N_{F^2}	0.993	0.476	0.254	”	1.00	1.00	1.001	1.0000	$9.71{\cdot}10^{-6}$
$D_{\text{abs}}, N_{\mathbb{E}}$	0.993	0.476	0.253	”	1.00	1.00	1.000	1.0000	$4.19{\cdot}10^{-3}$
D_{abs}, N_F	0.993	0.476	0.253	”	1.00	1.00	1.000	1.0000	$4.17{\cdot}10^{-3}$
Leading	0.990	0.479	0.250	”	1.00	1.00	1.006	0.9835	—
ca-GrQc									
Source	—	—	—	5242	14484	229867	2482738	48260	—
$D_{\text{sq}}, N_{\mathbb{E}}$	1.000	0.221	1.000	8192	3.52	2.74	1.028	0.0666	$9.14 \cdot 10^{5}$
$D_{\text{sq}}, N_{\mathbb{E}^2}$	1.000	0.733	0.000	”	4.30	29.82	355.084	0.9052	$2.53{\cdot}10^{0}$
D_{sq}, N_F	1.000	0.459	0.312	”	1.17	0.99	1.001	0.0107	$4.77{\cdot}10^{4}$
D_{sq}, N_{F^2}	1.000	0.467	0.279	”	1.06	0.92	1.035	0.0107	$9.89{\cdot}10^{-1}$
$D_{\text{abs}}, N_{\mathbb{E}}$	1.000	0.737	0.000	”	4.51	32.38	397.213	1.0000	$2.75{\cdot}10^{0}$
D_{abs}, N_F	1.000	0.469	0.267	”	1.00	0.87	1.000	0.0103	$1.12{\cdot}10^{0}$
Leading	1.000	0.488	0.229	”	1.00	1.00	1.405	0.0131	—
as20000102									
Source	—	—	—	6474	12572	2059364	$6.75{\cdot}10^{8}$	6584	—
$D_{\text{sq}}, N_{\mathbb{E}}$	1.000	0.722	0.000	8192	4.42	2.73	0.997	5.2222	$2.32{\cdot}10^{6}$
$D_{\text{sq}}, N_{\mathbb{E}^2}$	0.712	0.947	0.000	”	10.13	4.89	0.840	1.1082	$1.49{\cdot}10^{0}$
D_{sq}, N_F	1.000	0.722	0.000	”	4.40	2.71	0.989	5.1843	$6.39{\cdot}10^{6}$
D_{sq}, N_{F^2}	1.000	0.632	0.000	”	1.63	0.51	0.101	0.7029	$1.54{\cdot}10^{0}$
$D_{\text{abs}}, N_{\mathbb{E}}$	0.676	0.980	0.000	”	11.83	5.98	1.000	1.0000	$1.75{\cdot}10^{0}$
D_{abs}, N_F	1.000	0.648	0.000	”	1.95	0.68	0.152	1.0000	$2.12{\cdot}10^{0}$

Table 2. For three graphs, the fitted Kronecker parameters a, b, c for variations on the objective function (4.2). Subsequent columns show feature counts (vertices, edges, hairpins, tripins, triangles) for these parameters. The row labeled Source shows the actual network feature values F_{obs}. The other rows show $\mathbb{E}(F)/F_{\text{obs}}$. The objective column shows the value of the objective function at the minimizer.

All of the previously described networks have distinctly skewed degree distributions. That is, there are a few authors, actors, routers, or articles with a large number of links, despite the overall network having a small average degree. The final network we study is usroads, a highway-level network from the National Highway Planning Network (http://www.fhwa.dot.gov/planning/nhpn/), which does not have a highly skewed distribution. We include it as an example of a nearly planar network. It is also naturally undirected. The file is also available from the UFL repository.

In two of the experiments, we generate synthetic Kronecker networks. The algorithm to realize these networks is an explicit coin-flipping procedure instead of the more common ball-dropping method [Leskovec et al. 10]. For each cell i, j in the $\binom{2^r-1}{2}$ upper triangular portion, we first determine the log of the probability of a nonzero value in that cell, then generate a random coin flip with that probability as heads and record an edge when the coin comes up heads. This procedure is scalable because the full matrix of probabilities is never formed. It is also easily parallelizable. Our implementation uses pthreads to exploit multicore parallelism. It takes somewhat more work than the ball-dropping procedure, scaling as $O(r2^{2r})$ instead of $O(rm)$, where m is the number of balls dropped. Often $m \approx 2^{r+3}$, that is, eight balls per vertex [Groër et al. 11]. Each ball generates about one edge; see [Groër et al. 11] for a more thorough analysis. Coin-flipping preserves the exact Kronecker distribution, whereas ball-dropping is an approximation.

The experiments with these networks investigate (i) the difference in results from the various choices of D and N in the objective (4.2); (ii) the fitted parameters to the eight real-world networks; and (iii) the difference in fitted parameters when only three of the four graph features are used.

5.1. Objective Functions

The first study regards the choice of objective function. Of eight possible combinations of distance and normalization, we considered two to be unreasonable a priori. Here we investigate the other six pairs.

Table 2 shows the different parameters a, b, and c chosen by each objective function, as well as the expected feature counts for those parameters for three graphs: a single realization of a Kronecker graph with $a = 0.99$, $b = 0.48$, $c = 0.25$, the collaboration network ca-GrQc, and the infrastructure network as20000102. The rows labeled "Source" contain the actual feature counts in each network. The optimization algorithm to pick a, b, c uses the best objective value from three procedures. First, it tries 50 random starting points for the fmincon function in Matlab R2010b (a constrained optimizer based on an active set algorithm).

Then, it performs a grid search procedure with 100 equally spaced points in each dimension. Finally, it tries the leading-term-matching algorithm from Section 4.3 and considers those parameters.

The results in the table show that the choice of objective function does not make a difference when the graph fits the Kronecker model. However, it can make a large difference when the graph does not exactly fit, as in the ca-GrQc and as20000102 networks. Both of the objectives $D_{\mathrm{sq}}, N_{\mathbb{E}^2}$ and $D_{\mathrm{abs}}, N_{\mathbb{E}}$ produced distinctly different fits for these two networks, compared to the other objectives. These two fits seem to be primarily matching the number of triangles—almost to the exclusion of the other features. The other odd fit for the ca-GrQc graph comes from the $D_{\mathrm{sq}}, N_{\mathbb{E}}$ objective. This fit appears to be matching the tripin count and ignoring other features, something that also seems to be true for the as20000102 graph. Among the remaining fits for ca-GrQc, there is little difference among the fitted parameters and estimated features. The results are a bit different for as20000102. The fits for $D_{\mathrm{sq}}, N_{\mathbb{E}}$ and D_{sq}, N_F are almost identical and show a good match to the tripin count, but a poor match to the remaining features. The fits for D_{sq}, N_{F^2} and D_{abs}, N_F are similar, and deciding which is better seems like a matter of subjective preference. These observations held up under further experimentation, which we omit here in the interest of space.

Based on these results, either of the objectives D_{sq}, N_{F^2} and D_{abs}, N_F appears to be a robust choice when the model does not fit exactly. Due to the continuity of the D_{sq} function, the rest of our fits in this paper use the D_{sq}, N_{F^2} variation.

5.2. Parameters for Real-World Networks

For the eight networks previously described, we use the objective function (4.2) with D_{sq}, N_{F^2} to fit the parameters a, b, c. The results, along with the expected feature counts for the fitted parameters, are presented in Table 3. We show the minimizer for the three different strategies to optimize the objective described in the previous section: a direct minimization procedure, the grid-search procedure, and the leading-term-matching approach (Section 4.3). For each approach, the table also shows the time required for that algorithm and the value of the objective function at the minimizer.

In [Leskovec et al. 10], the authors provide the fitted parameters a, b, and c from their KronFit algorithm for the networks ca-GrQc, ca-HepTh, ca-HepPh, and as20000102. We include them in Table 3 for comparison. In all cases but one, the expected feature count using KronFit is farther from the observed feature count than the expectation under our moment-based fits. Sometimes it is much farther. There was a sole exception on a single feature. For the graph as20000102, KronFit gave a better estimate of the number of edges than our moment method gave; however,

Graph	Kron. Parameters			Graph/Expected Features						Time
Fit type	a	b	c	Vertices	Edges	Hairpins	Tripins	Tris.	Obj.	(sec.)
ca-GrQc										
Source	—	—	—	5242	14484	229867	2482738	48260	—	<0.05
Direct	1.000	0.467	0.279	8192	1.06	0.92	1.035	0.0107	0.989	1.0
Grid	1.000	0.470	0.270	”	1.03	0.91	1.060	0.0108	0.991	48.5
Leading	1.000	0.488	0.229	”	1.00	1.00	1.405	0.0131	1.138	<0.05
KronFit	0.999	0.245	0.691	”	0.84	0.20	0.029	0.0012	2.935	—
ca-HepPh										
Source	—	—	—	12008	118489	15278011	$1.28{\cdot}10^9$	3358499	—	1.9
Direct	1.000	0.669	0.101	16384	1.11	0.82	1.064	0.0164	1.015	0.8
Grid	1.000	0.670	0.100	”	1.12	0.84	1.091	0.0167	1.016	48.6
Leading	1.000	0.708	0.005	”	1.00	1.00	2.021	0.0196	2.004	<0.05
KronFit	0.999	0.437	0.484	”	0.69	0.10	0.014	0.0006	3.196	—
ca-HepTh										
Source	—	—	—	9877	25973	299356	2098335	28339	—	<0.05
Direct	1.000	0.401	0.379	16384	1.06	0.92	1.035	0.0112	0.989	0.8
Grid	1.000	0.400	0.380	”	1.05	0.90	1.001	0.0109	0.991	48.7
Leading	1.000	0.423	0.325	”	1.00	1.00	1.444	0.0140	1.169	<0.05
KronFit	0.999	0.271	0.587	”	0.74	0.25	0.073	0.0020	2.936	—
hollywood										
Source	—	—	—	1139905	56375711	$4.76{\cdot}10^{10}$	$3.24{\cdot}10^{13}$	$4.92{\cdot}10^9$	—	2946.1
Direct	1.000	0.623	0.186	2097152	1.13	0.76	1.070	0.0029	1.075	1.0
Grid	1.000	0.620	0.200	”	1.21	0.80	1.055	0.0030	1.083	48.6
Leading	1.000	0.662	0.095	”	1.00	1.00	2.670	0.0046	3.779	<0.05
as20000102										
Source	—	—	—	6474	12572	2059364	$6.75{\cdot}10^8$	6584	—	<0.05
Direct	1.000	0.632	0.000	8192	1.63	0.51	0.101	0.7029	1.541	0.8
Grid	1.000	0.630	0.000	”	1.60	0.49	0.096	0.6717	1.543	48.7
KronFit	0.987	0.571	0.049	”	0.99	0.17	0.018	0.1738	2.655	—

Table 3. The fitted Kronecker parameters for variations on the algorithm—direct, grid, leading, or KronFit [Leskovec et al. 10]—to minimize the objective function (4.2). Subsequent columns show the expected feature counts (vertices, edges, hairpins, tripins, triangles) for these parameters, then the value of the objective function. The row labeled Source shows the actual network features. The time column is either the time to compute the features on the original graph or the time for the algorithm to fit the parameters. (*Continued)*

Graph	Kron. Parameters			Graph/Expected Features						Time
Fit type	a	b	c	Vertices	Edges	Hairpins	Tripins	Tris.	Obj.	(sec.)
as-skitter										
Source	—	—	—	1696415	11095298	$1.60{\cdot}10^{10}$	$9.66{\cdot}10^{13}$	28769868	—	107.0
Direct	1.000	0.644	0.000	2097152	1.61	0.74	0.239	0.1384	1.755	0.7
Grid	1.000	0.640	0.000	”	1.48	0.65	0.199	0.1181	1.776	48.7
wiki-2005										
Source	—	—	—	1634989	18540603	$3.72{\cdot}10^{10}$	$3.72{\cdot}10^{14}$	44667105	—	378.9
Direct	1.000	0.674	0.000	2097152	1.64	0.79	0.211	0.2589	1.629	0.6
Grid	1.000	0.670	0.000	”	1.53	0.70	0.179	0.2246	1.646	48.5
usroads										
Source	—	—	—	126146	161950	292425	115885	4113	—	<0.05
Direct	1.000	0.070	1.000	131072	0.88	1.04	1.057	0.1177	0.798	1.0
Grid	1.000	0.070	1.000	”	0.87	1.03	1.012	0.1148	0.800	48.5

Table 3. Continued.

that came at the expense of lack of fit to the other features. In the previous section we discussed that fitting as20000102 involved a trade-off among which particular features were fit, since the graph seems not to fit the Kronecker model. KronFit appears to have focused on the edge count in this case.

KronFit typically underestimates the feature counts. The effect is severe for triangles. Kronecker random graphs commonly have many fewer triangles than the real-world graphs to which they are fit. Our moment-based estimators find parameters leading to many more triangles than the KronFit parameters do.

In fairness, we point out that our method is designed to match expected-to-observed feature counts, while KronFit fits by maximum likelihood. Therefore, the evaluation criterion is closer to the fitting criterion for us. But maximum likelihood ordinarily beats or matches the method of moments in large samples from parametric models; its mismatching criteria are more than compensated by superior statistical efficiency. The explanation here may involve maximum likelihood being less robust to lack of fit of the Kronecker model, or it may be that KronFit is not finding the MLE. Our evaluations of likelihood using the SNAP implementation of KronFit (http://snap.stanford.edu) yielded inconclusive results due to the stochastic nature of the likelihood computations. These computations are available in the online codes.

The results in Table 3 show small differences in the fits between the direct and grid algorithms, although the direct algorithm is much faster. The leading-term-matching algorithm, when it succeeds, generates similar Kronecker parameters, although with a distinctly worse objective value. The results from the KronFit algorithm differ and likely match the graph in another aspect.

Leading-term matching is tens of times faster than direct search and roughly one thousand times faster than grid search. But even the grid search takes under a minute in our examples, so the speed savings from the leading-term approach is of little benefit here. For the large graphs, the time to compute the network features dominates the time to fit the parameters, showing that this approach scales to large networks.

Overall, the results indicate that the Kronecker models tend not to be a good fit to the data. The model appears to have a considerable difference in at least one of the graph features. Usually, it is the number of triangles, which differs by up to two orders of magnitude for many of the collaboration networks.

5.3. Fitting Partial Sets of Features

The previous set of experiments illustrated that the Kronecker graphs may not simultaneously fit all four of the network features: edges, hairpins/wedges, tripins, and triangles. In Table 4, we examine the change in fits when using only three of the four network features in the summation in the objective (4.2). We take the set of parameters with the smallest objective among all the procedures investigated in the previous section. The results show small changes to the parameters and expected feature fits. Nonetheless, the minimizer remains mostly unchanged.

Table 4 provides a kind of cross-validated feature estimation, showing the accuracy of a feature's estimate when it is not included in the fitting. Apart from the exception noted earlier (the edge counts for as20000102), our moment-based estimates give closer matches to the source feature counts than KronFit provides, whether the moment being studied is part of the fitting process or not.

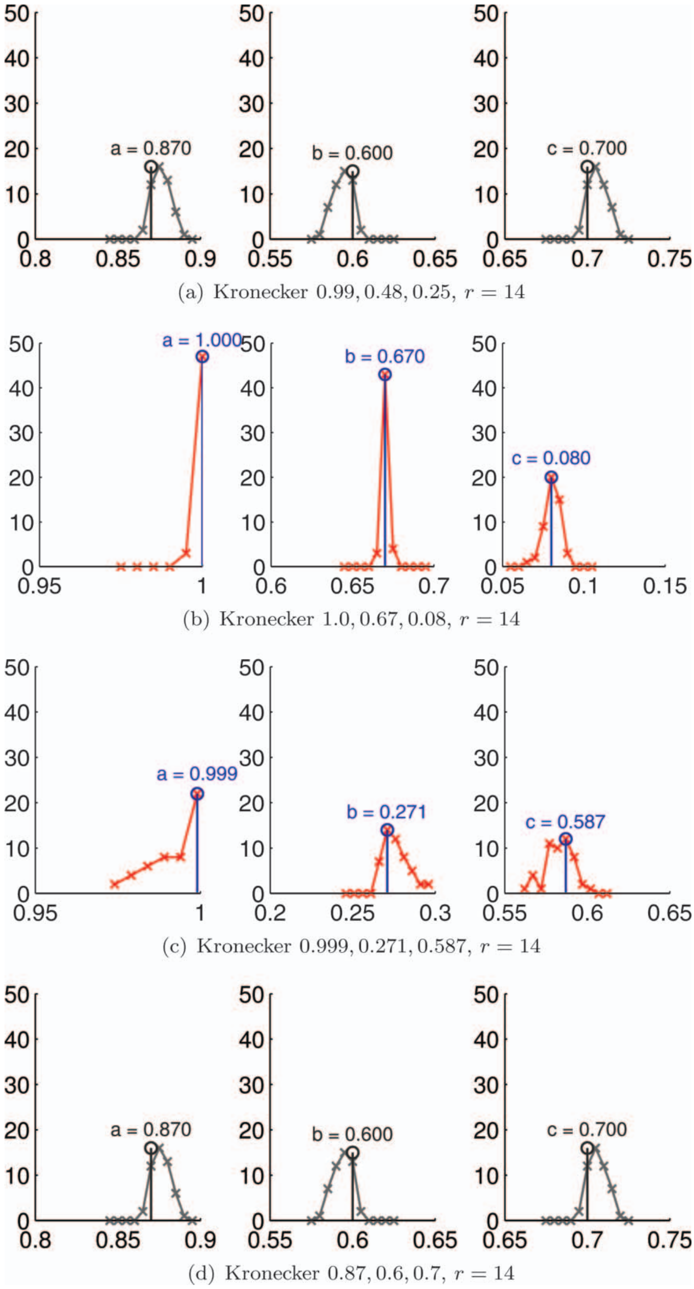
We see some examples in which leaving out one feature seems to improve the fitting of another. For instance, in three of the four graphs, leaving out the tripin count improved the match for triangles.

6. Synthetic Examples

The results from the previous section show that there can often be a large deviation in the expected moments of the best Kronecker fit. In this section, we investigate the accuracy of the fitting procedure when the graph is a realization of a stochastic Kronecker network.

Graph	Kroneck. Parameters			Graph/Expected Features						Time
Fit type	*a*	*b*	*c*	Vertices	Edges	Hairpins	Tripins	Tris.	Obj.	(sec.)
ca-GrQc										
Source	—	—	—	5242	14484	229867	2482738	48260	—	<0.05
All	1.000	0.467	0.279	8192	1.06	0.92	1.035	0.0107	0.989	54.8
KronFit	0.999	0.245	0.691	”	0.84	0.20	0.029	0.0012	2.935	—
-Edges	1.000	0.458	0.317	”	1.19	1.00	1.007	0.0108	0.978	53.9
-Hairpins	1.000	0.469	0.267	”	1.00	0.87	1.007	0.0103	0.980	53.9
-Tripins	1.000	0.493	0.216	”	0.99	1.02	1.536	0.0139	0.973	54.0
-Tris	1.000	0.467	0.279	”	1.06	0.92	1.029	0.0106	0.011	56.1
ca-HepPh										
Source	—	—	—	12008	118489	15278011	$1.28{\cdot}10^9$	3358499	—	1.9
All	1.000	0.669	0.101	16384	1.11	0.82	1.064	0.0164	1.015	54.1
KronFit	0.999	0.437	0.484	”	0.69	0.10	0.014	0.0006	3.196	—
-Edges	1.000	0.650	0.192	”	1.49	1.02	1.006	0.0201	0.960	57.2
-Hairpins	1.000	0.670	0.083	”	1.01	0.75	1.007	0.0146	0.971	57.2
-Tripins	1.000	0.709	0.005	”	1.01	1.02	2.065	0.0200	0.961	56.9
-Tris	1.000	0.669	0.099	”	1.10	0.82	1.058	0.0162	0.047	54.7
ca-HepTh										
Source	—	—	—	9877	25973	299356	2098335	28339	—	<0.05
All	1.000	0.401	0.379	16384	1.06	0.92	1.035	0.0112	0.989	57.4
KronFit	0.999	0.271	0.587	”	0.74	0.25	0.073	0.0020	2.936	—
-Edges	1.000	0.391	0.417	”	1.19	1.00	1.006	0.0114	0.977	56.5
-Hairpins	1.000	0.404	0.365	”	1.00	0.87	1.008	0.0108	0.979	57.1
-Tripins	1.000	0.431	0.308	”	0.98	1.03	1.623	0.0152	0.971	56.9
-Tris	1.000	0.401	0.379	”	1.06	0.92	1.028	0.0111	0.011	56.6
as20000102										
Source	—	—	—	6474	12572	2059364	$6.75{\cdot}10^8$	6584	—	<0.05
All	1.000	0.632	0.000	8192	1.63	0.51	0.101	0.7029	1.541	56.7
KronFit	0.987	0.571	0.049	”	0.99	0.17	0.018	0.1738	2.655	—
-Edges	0.935	0.720	0.000	”	3.04	1.12	0.235	1.0796	0.608	57.3
-Hairpins	1.000	0.621	0.000	”	1.44	0.41	0.077	0.5526	1.250	58.5
-Tripins	1.000	0.628	0.000	”	1.56	0.47	0.091	0.6400	0.723	56.7
-Tris	1.000	0.618	0.000	”	1.39	0.39	0.071	0.5137	1.392	57.1

Table 4. The change in fitted parameters when the objective function (4.2) considers only three of the four features. The row labeled “-Tris,” for instance, gives the fitted parameters when triangles are not included in (4.1). Rows labeled “source” again contain the actual graph features, and the rows labeled “all” show the parameters fitted to all four features. The columns are as in Table 3.



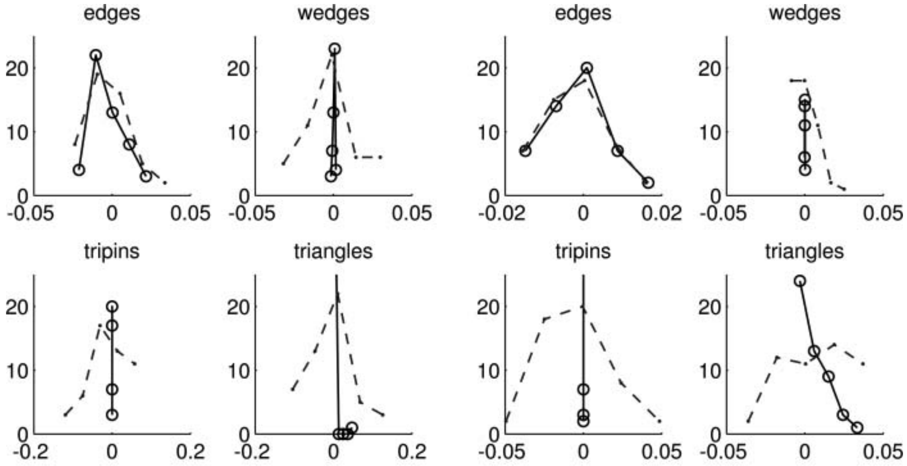
(a) Kronecker $0.99, 0.48, 0.25,\ r = 14$

(b) Kronecker $1.0, 0.67, 0.08,\ r = 14$

(c) Kronecker $0.999, 0.271, 0.587,\ r = 14$

(d) Kronecker $0.87, 0.6, 0.7,\ r = 14$

Figure 2. Histograms of fitted parameters to 50 realizations of a Kronecker graph with the parameters given in the caption (lines with x's) (color figure available online).



(a) Kronecker $0.99, 0.48, 0.25$, $r = 14$

(b) Kronecker $1.0, 0.67, 0.08$, $r = 14$

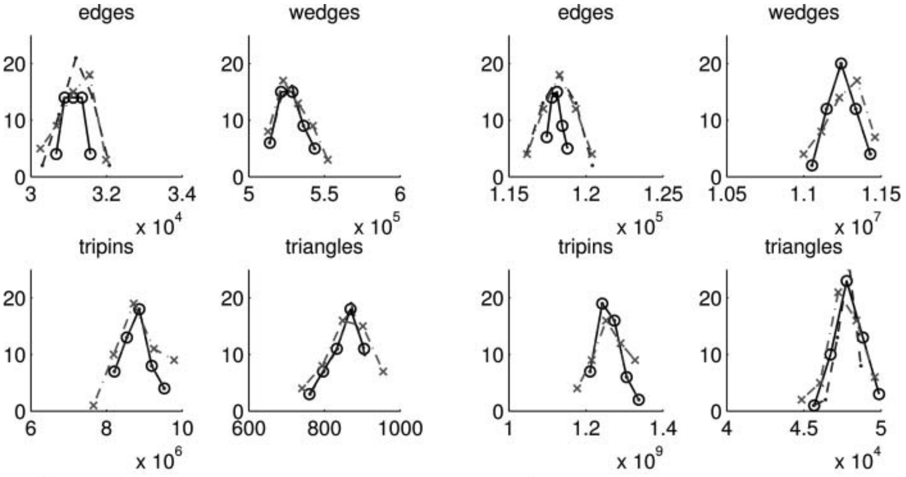
Figure 3. Histograms of the relative difference between the true graph feature and the fitted (solid) or regenerated (dashed line) feature. The relative difference is $(F_{\text{true}} - F_{\text{fit}})/F_{\text{true}}$.

For four sets of Kronecker parameters

$$
\begin{aligned}
(a, b, c) &= (0.99, 0.48, 0.25), \quad r = 14,\\
(a, b, c) &= (1.0, 0.67, 0.08), \quad r = 14,\\
(a, b, c) &= (0.999, 0.271, 0.587), \quad r = 14,\\
(a, b, c) &= (0.87, 0.6, 0.7), \quad r = 14,
\end{aligned}
$$

we generate 50 realizations of each Kronecker graph. For each realization, we compute a fit using the objective (4.2) with the choices D_{sq}, N_{F^2} and using the combination of approaches from the previous section. Figure 2 shows the distribution of fitted parameters to these 50 samples. For all four sets of parameters, the fitted results closely match the true values, with fairly small variation.

For these synthetic problems, we also study how the empirical and fitted features differ. Figure 3 shows the distribution of the relative difference between the expectation of the fitted Kronecker features and the actual feature of each realization. It also shows the difference between the original feature count and the feature count of a re-realization. In other words, generate a Kronecker graph, fit the parameters, and re-generate with the fitted parameters. The figures show that the fitted parameters closely match the realizations. A curious property is that the fitted triangle count is always smaller than the empirical count. The



(a) Kronecker $0.99, 0.48, 0.25$, $r = 14$

(b) Kronecker $1.00, 0.67, 0.08$, $r = 14$

Figure 4. Histograms of empirical features. The dashed line with no × marks shows the empirically measured features, and the solid line with circles shows the expected value of each feature given the fitted parameters. These lines are often on top of each other. The dashed line with × marks shows the features of a regenerated graph.

difference in the re-realization can be large, almost 20% in the case of tripins or triangles for the first set of Kronecker parameters.

Our final study is the distributions of the graph features given the Kronecker parameters, the expected features of the fitted parameters, and the graph features of a re-realized Kronecker graph. The plots in Figure 4 show that these distributions are all quite similar.

7. Conclusions

We have presented formulas for expected feature counts in Kronecker graphs and used them to generate a method-of-moments fitting strategy. We found that summing squared relative feature count errors was robust and easy to optimize. For graphs generated by the Kronecker model, our parameter and feature estimates closely match those of the fitted graph. For real-world graphs we often find that the fitted Kronecker model implies smaller feature counts (apart from edges) than are seen in the real graph. The moment estimators typically come closer to the counts than those from KronFit. The recent article [Moreno et al. 10] describes a way of inducing dependence into edge probabilities for Kronecker-like graphs that may increase their triangle density.

Acknowledgments. We thank Tamara Kolda, C. Seshadhri, and Ali Pinar for helpful discussions. This work was supported by DMS-0906056 of the National Science Foundation.

Professor Gleich's work was completed while he was at Sandia National Laboratories, a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[Boldi and Vigna 04] Paolo Boldi and Sebastiano Vigna. "The Webgraph Framework I: Compression Techniques." In *Proceedings of the 13th International Conference on the World Wide Web*, pp. 595–602. ACM Press, 2004.

[Boldi et al. 11] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. "Layered Label Propagation: A Multiresolution Coordinate-Free Ordering for Compressing Social Networks." In *Proceedings of the 20th WWW2011*, pp. 587–596, 2011.

[Constantine and Gleich 07] Paul G. Constantine and David F. Gleich. "Using Polynomial Chaos to Compute the Influence of Multiple Random Surfers in the PageRank Model." In *Proceedings of the 5th Workshop on Algorithms and Models for the Web Graph (WAW2007)*, edited by Anthony Bonato and Fan Chung Graham, Lecture Notes in Computer Science 4863, pp. 82–95. Springer, 2007.

[Groër et al. 11] Chris Groër, Blair D. Sullivan, and Steve Poole. "A Mathematical Analysis of the R-MAT Random Graph Generator." *Networks* 58:3 (2011), 159–170.

[Horn and Radcliffe 11] P. Horn and M. Radcliffe. "Giant Components in Kronecker Graphs." *Random Structures and Algorithms* 40:3 (2012), 385–397.

[Kolountzakis et al. 10] Mihail Kolountzakis, Gary Miller, Richard Peng, and Charalampos Tsourakakis. "Efficient Triangle Counting in Large Graphs via Degree-Based Vertex Partitioning." In *Algorithms and Models for the Web-Graph*, edited by Ravi Kumar and Dandapani Sivakumar, Lecture Notes in Computer Science 6516, pp. 15–24. Springer, 2010.

[Leskovec and Faloutsos 07] J. Leskovec and C. Faloutsos. "Scalable Modeling of Real Graphs Using Kronecker Multiplication." In *International Conference on Machine Learning (ICML)*, 2007.

[Leskovec et al. 05] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. "Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication." In *European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2005.

[Leskovec et al. 07] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. “Graph Evolution: Densification and Shrinking Diameters.” *ACM Trans. Knowl. Discov. Data* 1 (2007), 1–41.

[Leskovec et al. 10] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. “Kronecker Graphs: An Approach to Modeling Networks.” *Journal of Machine Learning Research* 11 (2010), 985–1042.

[Mahdian and Xu 07] M. Mahdian and Y. Xu. “Stochastic Kronecker Graphs.” In *Proceedings of the 5th Workshop on Algorithms and Models for the Web-Graph (WAW2007)*, pp. 179–186, 2007.

[Moreno et al. 10] S. Moreno, S. Kirshner, J. Neville, and S.V.N. Vishwanathan. “Tied Kronecker Product Graph Models to Capture Variance in Network Populations.” In *48th Annual Allerton Conference onCommunication, Control, and Computing (Allerton), 2010*, pp. 1137–1144, 2010.

[Palla et al. 10] G. Palla, L. Lovaász, and T. Vicsek. “Multifractal Network Generator.” *Proceedings of the National Academy of Sciences* 107:17 (2010), 7640–7645.

[Schank and Wagner 05] T. Schank and D. Wagner. “Finding, Counting, and Listing All Triangles in Large Graphs.” In *Workshop on Experimental and Efficient Algorithms (WEA)*, 2005.

[Seshadhri et al. 11] C. Seshadhri, A. Pinar, and T. Kolda. “An In-Depth Study of Stochastic Kronecker Graphs.” In *Proceedings of IEEE International Conference on Data Mining*, ICDM2011, 2011.

[Tsourakakis 08] C. E. Tsourakakis. “Fast Counting of Triangles in Large Real Networks without Counting: Algorithms and Laws.” In *Eighth IEEE International Conference on Data Mining, 2008. ICDM ’08.*, pp. 608–617, 2008.

David F. Gleich, Purdue University, Office 1207, Lawson Computer Science Building, 305 N. University Ave., West Lafayette, IN 47907, USA (dgleich@purdue.edu)

Art B. Owen, Stanford University, Department of Statistics, Sequoia Hall, Stanford, CA 94305, USA (owen@stanford.edu)