Kronecker Graph Generation with Ground Truth for 4-Cycles and Dense Structure in Bipartite Graphs

Trevor Steil†
School of Mathematics
University of Minnesota
Minneapolis, MN, USA
steil016@umn.edu

Scott McMillan‡
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA, USA
smcmillan@sei.cmu.edu

Geoffrey Sanders†, Roger Pearce†, Benjamin Priest†
Center for Applied Scientific Computing (CASC)
Lawrence Livermore National Laboratory (LLNL)
Livermore, CA, USA
{sanders29, pearce7, priest2}@llnl.gov

***Abstract*—We demonstrate nonstochastic Kronecker graph generators produce massive-scale bipartite graphs with ground truth global and local properties and discuss their use for validation of graph analytics. Given two small connected scale-free graphs with adjacency matrices A and B, their Kronecker product graph [1] has adjacency matrix $C = A \otimes B$. We first demonstrate that having one factor A non-bipartite (alternatively, adding all self loops to a bipartite A) with other factor B bipartite ensures $\mathcal{G}_C$ is bipartite and connected.**

Formulas for ground truth of many graph properties (including degree, diameter, and eccentricity) carry over directly from the general case presented in previous work [2], [3]. However, the analysis of higher-order structure and dense structure is different in bipartite graphs, as no odd-length cycles exist (including triangles) and the densest possible structures are bicliques. We derive formulas to give ground truth for 4-cycles (a.k.a. squares or butterflies) at every vertex and edge in $\mathcal{G}_C$. Additionally, we demonstrate that bipartite communities (dense vertex subsets) in the factors A, B yield dense bipartite communities in the Kronecker product C.

We additionally discuss interesting properties of Kronecker product graphs revealed by the formulas an their impact on using them as benchmarks with ground truth for various complex analytics. For example, for connected A and B of nontrivial size, $\mathcal{G}_C$ has 4-cycles at vertices/edges associated with vertices/edges in A and B that have none, making it difficult to generate graphs with ground truth bipartite generalizations of truss decomposition (e.g. the k-wing decomposition of [4]).

I. Introduction

Bipartite graphs are common in real-world relational data analysis applications, including text analysis (term-document matrices), machine learning with discrete features (entity-feature matrices), and recommender systems (user-rating matrices). Graph analytics are an important data analysis tool for bipartite graph datasets, and developing performant algorithms and their implementations for various graph computations is a vibrant research area. It is important for the research community to have a collection of large-to-massive-scale bipartite graph datasets to validate their algorithm development and compare performance with other techniques. Collections of open real-world graphs are extremely important for this (several are available [5]–[7]), but synthetic generated graphs play an important role as well, particularly when the datasets are so large that knowing the correct answer of a given graph computation is challenging. Here we propose using non-stochastic Kronecker graphs to efficiently generate massive graphs with ground truth of various challenging local and global graph statistics, while having several of the challenging aspects of real-world graphs (heavy-tail degree distribution, some dense community structure, et cetera).

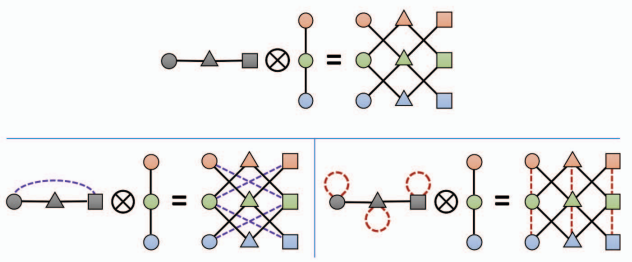


Fig. 1: Small examples of Kronecker products yielding bipartite graphs. (Top) Two bipartite connected factors yield a bipartite, yet disconnected graph, as discussed in §III-A. (Lower-Left) If both factors are connected, and one is non-bipartite, a connected bipartite graph is produced (additional edges are dashed and purple). (Lower-Right) Alternatively, if bipartite factors have self loops added to all vertices of one factor, the product is also bipartite and connected (additional edges are dashed and red). See Thms. 1 and 2.

For non-bipartite graphs a fundamental structure used in community analysis is the 3-cycle (a.k.a. triangle). Local counts of triangles at vertices and edges are important in several aspects of graph analysis, including clustering coefficients [8], truss decomposition [9], and edge reweighting for improved clustering [10], [11]. Previous work on non-stochastic Kronecker generators include several formulas for ground truth triangle counts [3], [12] and this technique has been used to validate global triangle counting on a trillion edge graph [13], a peta-scale graph computation.

For bipartite graphs, no odd-length cycle exists, so the *4-cycle* (a.k.a *square*, or *butterfly*) plays the role of the 3-cycle. Multiple proposals exist for extending clustering coefficient [14]–[16] and truss decomposition [4], [17] for bipartite graphs. Direct computation of local and global 4-

†This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and was approved for public release as LLNL-CONF-807167.

‡ This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center [DM20-0214].

978-1-7281-7445-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IPDPSW50202.2020.00052

cycle counts in sparse, real-world massive bipartite graphs $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is even more costly than 3-cycle counts in similar-sized non-bipartite graphs. A simple algorithm that runs a shortened breadth-first-search from each vertex $i \in \mathcal{V}$ into the second neighborhood of i and counts non-tree edges at each terminal vertex can count local vertex participation in 4-cycles and global counts of 4-cycles in $\mathcal{O}(|\mathcal{V}||\mathcal{E}|)$ for bipartite graphs. Several works describe how to improve direct computation for general (potentially non-bipartite) graphs [18], [19]. The compreshensive work [18] demonstrates worst-case bounds for cycle-detection algorithms with cycles of length 3 to 10, where the best bounds for detecting a length 4-cycle in a sparse graph are $\mathcal{O}(\mathcal{E}^{1.34})$ or $\mathcal{O}(\mathcal{E}\delta(\mathcal{G}))$, with $\delta(\mathcal{G})$ being the *degeneracy* of $\mathcal{G}$, an $\mathcal{O}(\mathcal{E}^{1/2})$ quantity.

Additionally, approximation techniques exist. The computational complexity makes graph generators that produce massive graphs with ground truth 4-cycle counts attractive for validating both direct and approximate computation techniques. A design criteria on these graph generators is that they yield graphs with similar challenges to real-world bipartite graphs, such as similarity with respect to size of maximum degree, heavy-tail degree distribution, dense structure, et cetera.

A proposed solution to this problem is to use nonstochastic Kronecker graphs as validation tools [20]. A *Kronecker graph* $\mathcal{G}_C$ has an adjacency matrix that is a Kronecker product [1], [21], [22] of two much smaller factors, $C = A \otimes B =$

$$\begin{pmatrix} A_{11}B & A_{12}B & \cdots & A_{1,n_A}B \\ A_{21}B & A_{22}B & \cdots & A_{2,n_A}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{n_A,1}B & A_{n_A,2}B & \cdots & A_{n_A,n_A}B \end{pmatrix},$$

For many graph statistics, these methods produce large-scale graphs with known ground truth statistics and properties [12].

Graph generation using nonstochastic Kronecker products can be contrasted with the stochastic Kronecker products used in the ubiquitous R-MAT generator [23]. R-MAT generators are used to produce the graphs used for various graph benchmarks, such as the Graph500 [24] and Graph Challenge [25], [26]. When using an R-MAT generator, exact graph properties cannot be determined until generation is complete, and their computation is expensive. After generation, large graphs take very large amounts of space if they are to be stored for reuse.

The use case for nonstochastic Kronecker generators is different from that of stochastic generators, and the former will not replace the latter. The nonstochastic Kronecker generators are appropriate for validation of algorithms and generation of graphs with certain properties at different scales. The generated graphs do have some peculiar properties, such as the lack of vertices with large prime degrees. Stochastic generators are appropriate for fast generation of graphs with certain properties, in expectation.

Previous work exists regarding scalable bipartite graph generators. A bipartite version of R-MAT exitsts [23], although the probability of generating high-order graph structure between medium-low degreee vertices is much too low to mimic many real-world bipartite graphs. In [27] authors developed a bipartite version of BTER (Block Two-Level Erd os-Renyi) that produces community structure in bipartite graphs and is fairly capable of matching degree-binned average of a type of bipartite clustering coefficient. When using these stochastic generators, some graph statistics are known in expectation. However, if an implementation of a complex graph statistic has a minor error (say a global count of 4-cycles is off by 1), it is difficult to know, without a competing implementation. Still it is difficult to know which implementation is correct, especially in massive graphs. Here, we focus on non-stochastic Kronecker generators, for which we demonstrate local and global ground truth of bipartite graph statistics is known, and researchers can use these generators and formulas to validate their novel algorithms and implementations.

Several previous works demonstrate useful Kronecker formulas and bounds for ground truth graph statistics, including degree distribution, triangle distribution, graph eccentricity, graph diameter, community structure, and eigenvalues [12], [20], [28], [29]. This paper extends these works by deriving efficient formulas for ground truth of 4-cycle counts and density in bipartite graphs. These quantities can be computed inexpensively and exactly for nonstochastic Kronecker products. In general, for a graph with $|\mathcal{E}_C|$ edges, suppose a desired graph analytic $f(C)$ costs $O(|\mathcal{E}_C|^p)$. If a simple Kronecker formula of the form

$$f(C) = \sum_s (g_s(A) \otimes h_s(B))$$

with a low number of terms exists, then a data structure requiring $O(|\mathcal{E}_C|^{p/2})$ storage can produce ground truth with $O(|f(C)| + |\mathcal{E}_C|^{p/2})$ cost. This means global scalar quantities (such as a global 4-cycle count) are computed sublinearly, in $O(|\mathcal{E}_C|^{p/2})$ time, and local quantities (such as 4-cycle counts at edges) are produced in linear time.

The linear algebraic ground truth formulas provided in this work lend themselves nicely to an implementation using GraphBLAS. The GraphBLAS Forum was formed in 2013 to standardize the mathematics and application programming interfaces (APIs) for performing graph computations in the language of linear algebra [29]. The mathematical specification, published in 2016 [30], was followed one year later by the first version of the GraphBLAS C API Specification [31]. The most recent release (version 1.3.0) of the GraphBLAS C API [32] included the Kronecker product operation that is used extensively throughout this derivation. In addition, GraphBLAS API also supports a non-blocking execution policy that would allow an implementation of the library to perform more optimizations of the code, through deferred/lazy evaluation, elimination of temporaries, and fusion of operations. With these optimizations, a relatively simple GraphBLAS code could be used to sample 4-cycle counts at edges and vertices without materializing the full Kronecker products to validate algorithms on massive graphs.

Our contributions are summarized as follows:

(a) We demonstrate that making one factor, A, non-bipartite (or, alternatively, bipartite A with all self loops) and B

bipartite with both factors connected ensures the product C is connected and bipartite.

(b) We extend the results in [3], [12] to derive Kronecker formulas for vertex and edge 4-cycle (a.k.a. square or butterfly) participation in the cases of one bipartite factor or self loops on every vertex in one of the factors. These results yield linear computation of ground truth local 4-cycles counts from a sublinear amount of memory.

(c) We derive scaling laws for bipartite edge clustering coefficients that demonstrate edge clustering coefficients are controllable.

(d) We derive Kronecker formulas and scaling laws for internal/external bipartite community edge counts and edge density, which are both controllable, under reasonable assumptions.

(e) We discuss several new advantages and disadvantages we have observed regarding using nonstochastic Kronecker bipartite graphs as various classes of benchmarks for massive-scale graph analytics.

II. Preliminaries

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a set of $n := |\mathcal{V}|$ vertices and $|\mathcal{E}|$ edges with pair-wise relationships between members of $\mathcal{V}$ of the form $(i, j) \in \mathcal{E}$, where $i, j \in \mathcal{V}$. We say $\mathcal{G}$ is *undirected* if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$ for every (i, j) (and $\mathcal{G}$ is *directed* if this doesn't hold for a single edge). An edge of the form $(i, i) \in \mathcal{E}$ is a *self loop*.

Let $\mathbb{B} = \{0, 1\}$. The matrix $A \in \mathbb{B}^{n\times n}$ is an *adjacency matrix* representing $\mathcal{G}$ if $A_{ij} = 1$ for each $(i, j) \in \mathcal{E}$ and $A_{ij} = 0$ for each $(i, j) \notin \mathcal{E}$. Given an adjacency matrix A, we use $\mathcal{G}_A$, $\mathcal{V}_A$, and $\mathcal{E}_A$, to represent the associated graph, vertices, and edges, respectively. An A associated with an undirected graph satisfies $A^t = A$. Additionally, we use a subscript A for many other symbols referring to properties of $\mathcal{G}_A$ (e.g. $n_A = |\mathcal{V}_A|$).

Def. 1. (Standard Matrix and Vector Objects) *Given $A \in \mathbb{R}^{n_A\times n_A}$, O_A is the* matrix of all zeros *and I_A is the* identity matrix*, both with the same size as A. Constant vectors $\mathbf{0}_A, \mathbf{1}_A \in \mathbb{R}^{n_A}$, are the* vector of all zeros*, and the* vector of all ones*. The cardinal vector $\mathbf{e}_i \in \mathbb{R}^{n_A}$ is a vector that is one in the i-th slot and zero elsewhere.*

Def. 2. (Walks and Trails) *A* walk *in $\mathcal{G}_A$ is a sequence of connected edges from $\mathcal{E}_A$, that possibly repeat. For example, (i_0, i_1), (i_1, i_0), (i_0, i_1), (i_1, i_2) is a walk with four hops. Powers of the adjacency matrix count walks in $\mathcal{G}_A$ of the length of the power. The number of walks from i to j of length h is*

$$W_A^{(h)}(i, j) := \mathbf{e}_i^t A^h \mathbf{e}_j.$$

The vector counting all walks of length h away from every vertex is $\mathbf{w}_A^{(h)} := A^h\mathbf{1}_A$. Note that $\mathbf{w}_A^{(1)} = \mathbf{d}_A$ is the vertex degree. A trail *is a walk that repeats no edges, forwards or backwards.*

Def. 3. (Closed Walks and Cycles) *A* closed walk *in $\mathcal{G}_A$ is a sequence of connected edges in $\mathcal{E}_A$ that start and end at the same vertex, and possibly repeat. For example, $(i_0, i_1), (i_1, i_2), (i_2, i_1), (i_1, i_0)$ is a closed walk with four hops. The number of closed walks of length h at vertex i is $W_A^{(h)}(i, i)$, or the i-th diagonal entry of A^h.*

Cycles *are closed walks that do not repeat or retrace any edges if (i_0, i_1) is in the cycle, then (i_0, i_1) or (i_1, i_0) are not present elsewhere in the cycle.*

Note that the number of 3-cycles at vertex i, or triangles t_i, is easily computed via $W_A^{(3)}(i, i) = 2t_i$, but the relationship is more complicated for length-4 closed walks and cycles, as discussed in §III-B.

A. Algebraic Properties of Kronecker Products

Matrices formed by Kronecker products are block structured and we define some convenience functions to write the index maps compactly. For a block-structured array with block-size n, we define functions that, for a given *global index* i, retrieve the *block number*, $\alpha_n(i)$, and the *intra-block index* $\beta_n(i)$.

$$\begin{aligned}\alpha_n(i) &= \lfloor (i-1)/n \rfloor + 1,\\ \beta_n(i) &= [(i-1)\%n] + 1.\end{aligned}$$

The inverse of $i \to (\alpha_n(i), \beta_n(i))$ is

$$\gamma_n(x, y) = (x-1)n + y,$$

in the sense that $i = \gamma_n(\alpha_n(i), \beta_n(i))$.

Def. 4. (Kronecker Product [1], [21], [22]) *Let $A \in \mathbb{R}^{m_A\times n_A}$ and $B \in \mathbb{R}^{m_B\times n_B}$. The* Kronecker Product *of A and B is $(A \otimes B) \in \mathbb{R}^{(m_A m_B)\times(n_A n_B)}$ and has entries*

$$(A \otimes B)_{pq} = \left(A_{\alpha_{m_B}(p),\alpha_{n_B}(q)}\right)\left(B_{\beta_{m_B}(p),\beta_{n_B}(q)}\right)$$

for $1 \le p \le (m_A m_B)$ and $1 \le q \le (n_A n_B)$, or, equivalently,

$$(A \otimes B)_{\gamma_{m_B}(i,k),\gamma_{n_B}(j,l)} = A_{ij}B_{kl},$$

for $1 \le i \le m_A, 1 \le j \le n_A, 1 \le k \le m_B$, and $1 \le l \le n_B$.

We reserve p and q to be row and column indices into $(A \otimes B)$, with i, j having similar roles for A and k, l for B. Kronecker row and column indices are always associated with pairs of factor indices via

Row Indices	Column Indices
$p = \gamma_{m_B}(i, k)$	$q = \gamma_{n_B}(j, l)$
$i = \alpha_{m_B}(p)$	$j = \alpha_{n_B}(q)$
$k = \beta_{m_B}(p)$	$l = \beta_{n_B}(p)$.

Def. 5. (Hadamard Product [22]) *Let $A_0, A_1 \in \mathbb{R}^{m\times n}$. The* Hadamard Product *of A_0 and A_1 is $(A_0 \circ A_1) \in \mathbb{R}^{m\times n}$, with*

$$(A_0 \circ A_1)_{ij} = (A_0)_{ij}(A_1)_{ij}$$

for $1 \le i \le m$ and $1 \le j \le n$.

We define some diagonal operators of square matrices in terms of the Hadamard product so it is transparent in Kronecker formulas we derive.

Def. 6. (Matrix Diagonal Operators and Self Loops) *Given $A \in \mathbb{R}^{n_A\times n_A}$, the matrix $D_A = I_A \circ A$ is the* diagonal entries

of A. The diagonal operator *is $diag(A) := (I_A \circ A)\mathbf{1}_A$, a vector in $\mathbb{R}^{n_A}$. Diagonal entries of D_A that are nonzero represent* self loops *in $\mathcal{G}_A$. When $D_A = I_A$, we say A has full self loops, and when $D_A = O_A$ we say A has no self loops.*

Throughout the derivations of Kronecker formulas, we utilize several of the algebraic properties of Kronecker and Hadamard products, as listed in Appendix A.

B. Self Loops in Kronecker Factors

Adding self loops to the Kronecker factors has been used previously to create denser structure when desired. For example, when self loops are added to A and B, a C with more triangles is produced. In this work, we also utilize self loops to ensure connectivity.

When both factors have self loops, the product C also has self loops. These must be removed via $[C - C \circ I_C]$ before using canonical linear algebra formulas for various graph statistics. For example if both factors have self loops $C = (A + I_A) \otimes (B + I_B)$, then the degree distribution vector is

$$\mathbf{d}_C = [C - C \circ I_C]\mathbf{1}_C,$$

and the triangle distribution vector is

$$\mathbf{t}_C = \frac{1}{2} diag\left([C - C \circ I_C]^3\right).$$

Thus, Kronecker formulas with self loops in both factors contain many more terms, as demonstrated in [3], [12].

On the other hand, when one factor of $C = M \otimes B$ has no self loops then the Kronecker product is an adjacency matrix with no self loops, or $B \circ I_B = O_B$ implies $(M \otimes B) \circ I_{M \otimes B} =$

$$(M \circ I_M) \otimes (B \circ I_B) = (M \circ I_M) \otimes O_B = O_{M \otimes B}.$$

This implies there are no self loops to remove before doing combinatoric computation via linear algebra, or $C - C \circ I_C = C$. Thus, we use no self loops in at least one factor in this work as some of the linear algebra formulas involve Kronecker products of pairs of 4-th matrix powers.

This design choice is important to limit the combinatorial complexity within deriving the formulas. If the input factors both had all self loops (e.g. $(A + I_A)$ and $(B + I_B)$), this would yield up to 25 terms in the derivation of formulas. If the diagonals of the factors had self loops added to only some of the vertices, then the off-diagonals do not commute with the diagonals, and there would be up to 256 terms in the derivations!

III. Bipartite Graphs

Def. 7. Bipartite Graph *A* bipartite *graph $\mathcal{G}_A(\mathcal{V}_A, \mathcal{E}_A)$ is one whose vertices can be grouped into two disjoint sets $\mathcal{U}_A \cup \mathcal{W}_A = \mathcal{V}_A$ such that no edge exists within either group, or $i, j \in \mathcal{U}_A$ implies $(i, j) \notin \mathcal{E}_A$, and $i', j' \in \mathcal{W}_A$ implies $(i', j') \notin \mathcal{E}_A$. Equivalently, bipartite graphs have no odd-length cycles. By ordering $\mathcal{U}_A$ before $\mathcal{W}_A$, the adjacency matrix of a bipartite graph is block anti-diagonal, or*

$$A = \begin{bmatrix} O_{|\mathcal{U}_A|} & X_A \\ Y_A^t & O_{|\mathcal{W}_A|} \end{bmatrix},$$

where $X_A, Y_A \in \mathbb{B}^{|\mathcal{U}_A| \times |\mathcal{W}_A|}$. If A is additionally undirected, then $Y_A = X_A$.

We assume factors A and B (and in turn, product C) are undirected throughout the rest of this section.

If one factor (say B, without loss of generality) is bipartite, then any Kronecker product graph with adjacency matrix $C = M \otimes B$ is also bipartite. This is seen by simply considering the two bipartite vertex sets $\mathcal{U}_B$ and $\mathcal{W}_B$ and the vertices in $\mathcal{V}_C$ associated with them. For p associated with $k \in \mathcal{U}_B$ and p' associated with $k' \in \mathcal{U}_B$, edge $(p, p') \notin \mathcal{E}_C$ because (k, k') cannot be in $\mathcal{E}_B$. The situation is the same for pairs of vertices associated with $\mathcal{W}_B$.

A. Connectivity of Bipartite Kronecker Graphs

A graph $\mathcal{G}_A$ is *connected* if for each vertex pair $i, j \in \mathcal{V}_A$ there exists a walk of edges from i to j. Algebraically, this means that for each i, j there exists a number of hops h such that the $(i, j)-$th entry of A^h is non-zero, and $hops_A(i, j)$ is defined as the smallest such number. We discuss assumptions we make on A and B to ensure $\mathcal{G}_C$ is connected, as it is often desired when benchmarking.

It is well known that if $C = A \otimes B$ is the Kronecker product of two connected bipartite factors with no self-loops, A, B, then C is disconnected [1]. This is seen by considering that there are four disjoint subsets of $\mathcal{V}_C$ given by the direct products (denoted $\oplus$) of the bipartite sets $\mathcal{U}_A, \mathcal{W}_A, \mathcal{U}_B$, and $\mathcal{W}_B$. Equivalently, $\mathcal{V}_C =$

$$\Big\{\mathcal{U}_A \oplus \mathcal{U}_B\Big\} \cup \Big\{\mathcal{U}_A \oplus \mathcal{W}_B\Big\} \cup \Big\{\mathcal{W}_A \oplus \mathcal{U}_B\Big\} \cup \Big\{\mathcal{W}_A \oplus \mathcal{W}_B\Big\}.$$

Consider a $(p, q) \in \mathcal{E}_C$ corresponding to $(i, j) \in \mathcal{E}_A$ and $(k, l) \in \mathcal{E}_B$. Without loss of generality, assume $i \in \mathcal{U}_A$, then $j \in \mathcal{W}_A$ because A is bipartite. In the first case, let $k \in \mathcal{U}_B$, meaning $l \in \mathcal{W}_B$ because B is also bipartite. This yields (p, q) connecting $\{\mathcal{U}_A \oplus \mathcal{U}_B\}$ to $\{\mathcal{W}_A \oplus \mathcal{W}_B\}$. In the second case, let $k \in \mathcal{W}_B$, meaning $l \in \mathcal{U}_B$. This yields (p, q) connecting $\{\mathcal{U}_A \oplus \mathcal{W}_B\}$ to $\{\mathcal{W}_A \oplus \mathcal{U}_B\}$. There are no edges that connect these four sets in any other way. For this reason, we make two sets of assumptions that ensure connectivity but offer different properties.

Assump. 1. (Factors of Bipartite Kronecker Graphs)

(i) *Assume factor A is non-bipartite, undirected, and connected, and factor B is bipartite, undirected, and connected. Let*

$$C = A \otimes B.$$

(ii) *Assume that A is bipartite, undirected, connected, and has all self loops added, and B is bipartite, undirected, and connected. Let*

$$C = (A + I_A) \otimes B.$$

Assump. 1(i) yields the simplest Kronecker formulas, but is not composed of two bipartite graphs, which may be less ideal when factors are desired to match a real-world bipartite graph of interest. Assump. 1(ii) addresses this by using two true

bipartite factors, with a relatively simple set of formulas. As discussed at the end of the previous subsection, Assump. 1(i) and (ii) both imply C is bipartite, as factor B is bipartite. We prove connectivity of $\mathcal{G}_C$ in both of these cases.

Thm. 1. (Connectedness with a Non-Bipartite Factor) *Assume A is non-bipartite, undirected, and connected, and B is bipartite, undirected, and connected. If $C = A \otimes B$, then $\mathcal{G}_C$ is connected.*

Proof. For $p, q \in \mathcal{V}_C$, we see $W_C^{(h)}(p,q) =$

$$\mathbf{e}_p^t C^h \mathbf{e}_q = (\mathbf{e}_i^t A^h \mathbf{e}_j)(\mathbf{e}_k^t B^h \mathbf{e}_l) = W_A^{(h)}(i,j) \cdot W_B^{(h)}(k,l).$$

We need to show that there exists an h such that $W_A^{(h)}(i,j)$ and $W_B^{(h)}(k,l)$ are mutually nonzero. Factor A is non-bipartite and undirected, so there exists at least one odd-length cycle $\mathcal{C} \in \mathcal{V}_A$. Let s be a vertex involved in the cycle $\mathcal{C}$. Factor A is also fully connected, so a walk from i to j by way of s exists. Now a family of walks $\mathcal{S}_a$ of both even and odd length exist, by starting at i, heading to s, going around the cycle $a \in \mathbb{N}$ times, and then heading to j. The length is

$$|\mathcal{S}_a| = hops_A(i,s) + hops_A(s,j) + a|\mathcal{C}|. \tag{1}$$

Let $h^* = \max(hops_A(i,j), hops_B(k,l))$ where $hops_A(i,j)$ (and $hops_B(k,l)$) is the minimum hop distance from i to j in $\mathcal{G}_A$ (and k to l in $\mathcal{G}_B$). By Eqn. (1), there exists both an h_{odd} and h_{even} such that $W_A^{(h_{odd})}(i,j)$, $W_A^{(h_{even})}(i,j) > 0$ and $h_{odd}, h_{even} \geq h^*$.

Factor B is bipartite and undirected. If $k, l \in \mathcal{U}_B$ (or $k, l \in \mathcal{W}_B$) then $hops(k,l)$ is even. Otherwise, $k \in \mathcal{U}_B$ and $l \in \mathcal{W}_B$ (or $k \in \mathcal{W}_B$ and $l \in \mathcal{U}_B$) and $hops(k,l)$ is odd. Because a shortest path from k to l can be augmented into walks by traversing any edge incident to l back and forth $b \in \mathbb{N}$ times, another family of walks $\mathcal{T}_b$ exists with lengths

$$|\mathcal{T}_b| = hops(k,l) + b \cdot 2. \tag{2}$$

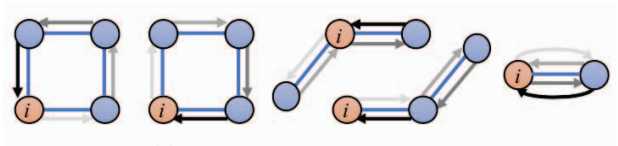
This means $W_B^{(h)}(k,l) > 0$ for every other integer greater or equal to $hops_B(k,l)$. As there exists $h_{odd}, h_{even} \geq h^*$ from above, we can pick a, b so $|\mathcal{S}_a| = |\mathcal{T}_b| \geq h^*$. Thus, $W_C^{(h)}(p,q) > 0$ for $h = |\mathcal{T}_b|$.

□

For the rest of this subsection we add all self loops on a bipartite factor A. Consider counting *lazy walks* by taking powers of $(A + I_A)$, which counts walks that also potentially wait at each hop, effectively removing any periodicity. The number of lazy walks with h stages (lazy walks up to length h, counting all potential wait sequences), is

$$\begin{aligned} W_{(A+I_A)}^{(h)}(i,j) &:= \mathbf{e}_i^t (A + I_A)^h \mathbf{e}_j \\ &= \sum_{r=0}^{h} \binom{h}{r} \mathbf{e}_i^t A^r \mathbf{e}_j = \sum_{r=0}^{h} \binom{h}{r} W_A^{(r)}(i,j), \end{aligned}$$

Note that $W_{(A+I_A)}^{(h)}(i,j) > 0$ for $h \geq hops_A(i,j)$ because it is the sum of non-negative terms containing $W_A^{(r)}(i,j) > 0$ for $r = hops_A(i,j)$. We use this observation to obtain the following result.



$$W_A^{(4)}(i,i) = 2s_i + d_i^2 + \sum_{j \in \mathcal{N}_i} d_j - d_i$$

Fig. 2: The number of length-4 closed walks that begin and end at vertex i, $W_A^{(4)}(i,i)$, double counts local 4-cycles at vertices s_i (once for each traversal direction), counts d_i^2 wedges centered at i, counts $\sum_{j \in \mathcal{N}_i} d_j$ wedges centered at vertices in $\mathcal{N}_i$. Both types of wedge represent (and thus double count) walks that go back and forth twice between i and every $j \in \mathcal{N}_i$, which is accounted for by subtracting d_i.

Thm. 2. (Connectedness with Self-Loops in One Factor) *Assume A and B are bipartite, undirected, and connected. Let $C = (A + I_A) \otimes B$. Then $\mathcal{G}_C$ is connected.*

Proof.

$$\begin{aligned} W_C^{(h)}(p,q) &= \mathbf{e}_p^t C^h \mathbf{e}_q = [\mathbf{e}_i^t (A + I_A)^h \mathbf{e}_j](\mathbf{e}_k^t B^h \mathbf{e}_l) \\ &= W_{(A+I_A)}^{(h)}(i,j) \cdot W_B^{(h)}(k,l) \end{aligned}$$

Let $h^* = \max(hops_A(i,j), hops_B(k,l))$. We showed that $W_{(A+I_A)}^{(h)}(i,j)$ is positive for any $h \geq h^*$, so we only need to show $W_B^{(h)}(k,l)$ is positive for some $h > h^*$. This is demonstrated simply by Eqn. (2). □

B. Kronecker Formulas for 4-Cycles

Here we derive Kronecker formulas for 4-cycles that are similar to those previously derived for 3-cycles [3], [12]. For both sets of assumptions 1(i) and (ii), we give formulas for local participation counts at vertices in §III-B1 and for local participation counts at edges in §III-B2. Then we show a relation for bipartite clustering coefficients of edges in §III-B3.

1) Vertex Participation in 4-Cycles:

Def. 8. (4-Cycles at Vertices) *If A has no self loops, then $\mathbf{s}_A \in \mathbb{R}^{n_A}$ is the vector storing the number of 4-cycles each vertex participates in,*

$$\mathbf{s}_A := \frac{1}{2}\left(diag(A^4) - \mathbf{d}_A \circ \mathbf{d}_A - \mathbf{w}_A^{(2)} + \mathbf{d}_A\right).$$

A point-wise formula is listed and explained in Fig. 2.

Thm. 3. (Vertex 4-Cycles w/o Self Loops) *Let A be nonbipartite, undirected, and connected and let B be bipartite and connected. Let $C = A \otimes B$, then $\mathbf{s}_C =$*

$$\begin{aligned} \tfrac{1}{2}\Big[&\left(2\mathbf{s}_A + \mathbf{d}_A^2 + \mathbf{w}_A^{(2)} - \mathbf{d}_A\right) \otimes \left(2\mathbf{s}_B + \mathbf{d}_B^2 + \mathbf{w}_B^{(2)} - \mathbf{d}_B\right) \\ &- \mathbf{d}_A^2 \otimes \mathbf{d}_B^2 - \mathbf{w}_A^{(2)} \otimes \mathbf{w}_B^{(2)} + \mathbf{d}_A \otimes \mathbf{d}_B\Big]. \end{aligned}$$

Proof.

$$\mathbf{s}_C = \frac{1}{2}\left(diag(C^4) - C\mathbf{1}_C \circ C\mathbf{1}_C - C^2\mathbf{1}_C + C\mathbf{1}_C\right)$$

Expand each term, $diag(C^4) = diag(A^4) \otimes diag(B^4) =$

$$\left(2\mathbf{s}_A + \mathbf{d}_A^2 + \mathbf{w}_A^{(2)} - \mathbf{d}_A\right) \otimes \left(2\mathbf{s}_B + \mathbf{d}_B^2 + \mathbf{w}_B^{(2)} - \mathbf{d}_B\right),$$

$$\begin{aligned} C\mathbf{1}_C &= (A \otimes B)(\mathbf{1}_A \otimes \mathbf{1}_B) = \mathbf{d}_A \otimes \mathbf{d}_B, \\ C^2\mathbf{1}_C &= (A^2\mathbf{1}_A) \otimes (B^2\mathbf{1}_B) = \mathbf{w}_A^{(2)} \otimes \mathbf{w}_B^{(2)}, \\ C\mathbf{1}_C \circ C\mathbf{1}_C &= (A\mathbf{1}_A \circ A\mathbf{1}_A) \otimes (B\mathbf{1}_B \circ B\mathbf{1}_B) \\ &= \mathbf{d}_A^2 \otimes \mathbf{d}_B^2. \end{aligned}$$

Combining and simplifying yields the final result. □

Given the formulas in [12] and the discussion in the beginning of §III, it is fairly easy to create Kronecker product graphs with no 3-cycles (in certain regions or globally). Moreover, it is possible to create Kronecker product graphs that have a ground truth truss decomposition. The situation is entirely different with 4-cycles.

Rem. 1. (Products Always Have 4-Cycles) *We note that it is difficult to create non-trivial Kronecker graphs with no 4-cycles, should one desire such a graph (see Fig. 1 for simple demonstrations).*

Assume A and B have no 4-cycles. we see that in this case

$$\begin{aligned} \mathbf{s}_C = \frac{1}{2}\Big[&\left(\mathbf{d}_A^2 + \mathbf{w}_A^{(2)} - \mathbf{d}_A\right) \otimes \left(\mathbf{d}_B^2 + \mathbf{w}_B^{(2)} - \mathbf{d}_B\right) \\ &- \mathbf{d}_A^2 \otimes \mathbf{d}_B^2 - \mathbf{w}_A^{(2)} \otimes \mathbf{w}_B^{(2)} + \mathbf{d}_A \otimes \mathbf{d}_B\Big]. \end{aligned}$$

If A and B both have a vertex with degree 2 or greater, there will be 4-cycles for $C = A \otimes B$. The only graphs that have all degree 1 vertices are collections of disjoint edges, which is an extremely limiting constraint.

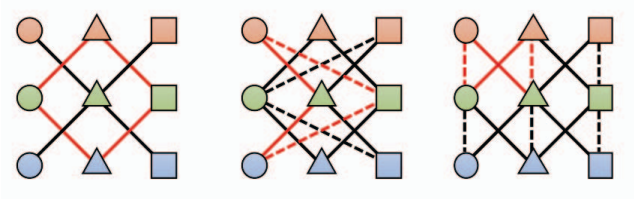
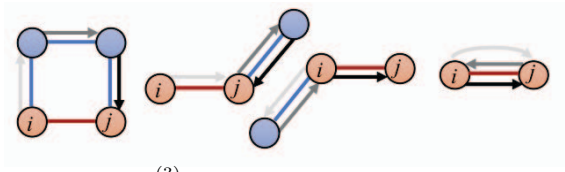


Fig. 3: Various types of 4-cycles in the Kronecker product graphs in the examples introduced in Fig. 1. Some of the 4-cycles are labelled with red edges.

Thm. 4. (Vertex 4-Cycles w/ All Self Loops in one Factor) *Let A and B be bipartite, undirected, and connected. Let $C = (A + I_A) \otimes B$, then*

$$\begin{aligned} \mathbf{s}_C = \tfrac{1}{2}\Big[&(2\mathbf{s}_A + \mathbf{d}_A^2 + \mathbf{w}_A^{(2)} + 5\mathbf{d}_A + \mathbf{1}_A) \otimes \\ &(2\mathbf{s}_B + \mathbf{d}_B^2 + \mathbf{w}_B^{(2)} - \mathbf{d}_B) \\ &-(\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{d}_B \\ &-(\mathbf{w}_A^{(2)} + 2\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{w}_B^{(2)} \\ &+(\mathbf{d}_A^2 + 2\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{d}_B^2\Big]. \end{aligned}$$



$$W_A^{(3)}(i,j) = \Diamond_{ij} + d_i + d_j - 1$$

Fig. 4: The number of length-3 walks that start vertex i and end at vertex $j \in \mathcal{N}_i$, $W_A^{(3)}(i,j)$, counts local 4-cycles at edges $\Diamond_{ij}$, and counts d_i and d_j wedges centered at i and j, respectively. Both types of wedge represent (and thus double count) the walk that goes back and forth between i and j, which is accounted for by subtracting 1.

We give a point-wise version of the formula for $p \in \mathcal{V}_C$ in terms of the associated vertices $i \in \mathcal{V}_A$ and $j \in \mathcal{V}_B$, $s_p =$

$$\begin{aligned} \tfrac{1}{2}\Big[&\left(2s_i + d_i^2 + w_i^{(2)} + 5d_i + 1\right)\left(2s_k + d_k^2 + w_k^{(2)} - d_k\right) \\ &-(d_i+1)d_k - \left(w_i^{(2)} + 2d_i + 1\right) w_k^{(2)} + (d_i+1)^2 d_k^2\Big]. \end{aligned}$$

Proof.

$$\mathbf{s}_C = \frac{1}{2}\left(diag(C^4) - C\mathbf{1}_C \circ C\mathbf{1}_C - C^2\mathbf{1}_C + C\mathbf{1}_C\right)$$

First note that $diag(A^3) = \mathbf{0}_A$ because A is bipartite, $diag(A^2) = \mathbf{d}_A$, and $diag(A) = \mathbf{0}_A$. Then, $diag(C^4) =$

$$\begin{aligned} &diag(A^4 + 4A^3 + 6A^2 + 4A + I_A) \otimes diag(B^4) \quad = \\ &\quad diag(A^4 + 6A^2 + I_A) \otimes diag(B^4) \quad = \\ &\quad (2\mathbf{s}_A + \mathbf{d}_A^2 + \mathbf{w}_A^{(2)} + 5\mathbf{d}_A + \mathbf{1}_A) \otimes \\ &\qquad (2\mathbf{s}_B + \mathbf{d}_B^2 + \mathbf{w}_B^{(2)} - \mathbf{d}_B). \end{aligned}$$

$$\begin{aligned} C\mathbf{1}_C &= [(A + I_A) \otimes B](\mathbf{1}_A \otimes \mathbf{1}_B) \\ &= (\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{d}_B. \end{aligned}$$

$$\begin{aligned} C^2\mathbf{1}_C &= [A^2\mathbf{1}_A + 2A\mathbf{1}_A + \mathbf{1}_A] \otimes (B^2\mathbf{1}_B) \\ &= (\mathbf{w}_A^{(2)} + 2\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{w}_B^{(2)}. \end{aligned}$$

$$\begin{aligned} &C\mathbf{1}_C \circ C\mathbf{1}_C = \\ &\quad = [A\mathbf{1}_A \circ A\mathbf{1}_A + 2A\mathbf{1}_A + \mathbf{1}_A] \otimes [B\mathbf{1}_B \circ B\mathbf{1}_B] \\ &\quad = (\mathbf{d}_A^2 + 2\mathbf{d}_A + \mathbf{1}_A) \otimes \mathbf{d}_B^2. \end{aligned}$$

□

2) Edge Participation:

Def. 9. (4-Cycles at Edges) *If A has no self loops, then $\Diamond_A \in \mathbb{R}^{n_A \times n_A}$ is the sparse matrix storing the number of 4-cycles each edge participates in,*

$$\Diamond_A = A^3 \circ A - (\mathbf{d}_A\mathbf{1}_A^t + \mathbf{1}_A\mathbf{d}_A^t) \circ A + A$$

A point-wise formula is listed and explained in Fig. 4.

Note the following relation between edge and vertex participation counts exists:

$$\mathbf{s}_A = \frac{1}{2}\Diamond_A\mathbf{1}_A.$$

Thm. 5. (Edge 4-Cycles w/o Self Loops) *Let A be non-bipartite, undirected, connected and B be bipartite, undirected, and connected. Let $C = A \otimes B$, then $\Diamond_C =$*

$$C + [\Diamond_A + (\mathbf{d}_A\mathbf{1}_A^t + \mathbf{1}_A\mathbf{d}_A^t) \circ A - A] \otimes [\Diamond_B + (\mathbf{d}_B\mathbf{1}_B^t + \mathbf{1}_B\mathbf{d}_B^t) \circ B - B] - [\mathbf{d}_A\mathbf{1}_A^t \circ A] \otimes [\mathbf{d}_B\mathbf{1}_B^t \circ B] - [\mathbf{1}_A\mathbf{d}_A^t \circ A] \otimes [\mathbf{1}_B\mathbf{d}_B^t \circ B].$$

We give a point-wise version of the formula for $(p, q) \in \mathcal{E}_C$ in terms of the associated edges $(i, j) \in \mathcal{E}_A$ and $(k, l) \in \mathcal{E}_B$,

$$\begin{aligned}\Diamond_{pq} &= \Diamond_{ij}\Diamond_{kl} + \Diamond_{ij}(d_k + d_l - 1) + (d_i + d_j - 1)\Diamond_{kl} \\ &\quad + d_i d_l - d_i - d_l + d_j d_k - d_j - d_k.\end{aligned}$$

Proof.

$$\Diamond_C = C^3 \circ C - (\mathbf{d}_C\mathbf{1}_C^t + \mathbf{1}_C\mathbf{d}_C^t) \circ C + C$$

$$(\mathbf{d}_C\mathbf{1}_C^t + \mathbf{1}_C\mathbf{d}_C^t) \circ C = [\mathbf{d}_A\mathbf{1}_A^t \circ A] \otimes [\mathbf{d}_B\mathbf{1}_B^t \circ B] + [\mathbf{1}_A\mathbf{d}_A^t \circ A] \otimes [\mathbf{1}_B\mathbf{d}_B^t \circ B].$$

$$\begin{aligned}C^3 \circ C &= (A^3 \circ A) \otimes (B^3 \circ B) \\ &= [\Diamond_A + (\mathbf{d}_A\mathbf{1}_A^t + \mathbf{1}_A\mathbf{d}_A^t) \circ A - A] \otimes \\ &\quad [\Diamond_B + (\mathbf{d}_B\mathbf{1}_B^t + \mathbf{1}_B\mathbf{d}_B^t) \circ B - B].\end{aligned}$$

The point-wise formula is given by expanding

$$\Diamond_{pq} = 1 + (\Diamond_{ij} + d_i + d_j - 1)(\Diamond_{kl} + d_k + d_l - 1) - d_i d_k - d_j d_l$$

into 19 terms and cancelling out 6 terms, then recombining. □

3) Clustering Coefficients: Clustering coefficients are useful local graph statistics for many applications. Often, graphs with a high amount of similar vertices have more triangles within sets of high affinity, and edge and vertex clustering coefficients (ratios of number of actual triangles to the maximum possible given the vertex degree(s)) are used to gauge local clustering [8].

For bipartite graphs, there are no triangles, and a generalization must be made. There are many notions of bipartite clustering coefficient proposed in the literature [14]–[16] that involve local counts of 4-cycles. Typically, the vertex clustering coefficient score of vertex i involves the degree and 4-cycle statistics of the vertices in $\mathcal{N}_i$ in addition to those statistics at i. For example, a vertex i with $d_i = 2$ could be involved in up to $min_{j \in \mathcal{N}_i}(d_j - 1)$ 4-cycles, which could potentially be a large number, independent of d_i.

In contrast, the number of 4-cycles an edge is possibly contained in is at most $(d_i - 1)(d_j - 1)$, or all vertices in $\mathcal{N}_i \setminus \{j\}$ connected to all vertices in $\mathcal{N}_j \setminus \{i\}$. In bipartite graphs, the number of triangles $\Delta_{ij} = 0$ and the overlap of these sets is empty, $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$. This edge-wise notion of clustering coefficient is related to the global bipartite clustering coefficient from [14] defined as a local coefficient and deftly named *metamorphosis coefficient* in [27].

Def. 10. (Bipartite Edge Clustering Coefficient [27])

$$\Gamma_A(i, j) = \frac{\Diamond_{ij}}{(d_i - 1)(d_j - 1)}$$

We derive a simple lower bound demonstrating that the edge clustering coefficient is controlled from below.

Thm. 6. (Scaling Law for Bipartite Clustering Coefficient) *Let A be non-bipartite and connected and B be bipartite and connected, and let $C = A \otimes B$. Let $p \in \mathcal{V}_C$ be associated with $i \in \mathcal{V}_A$ and $k \in \mathcal{V}_B$, and $q \in \mathcal{V}_C$ be associated with $j \in \mathcal{V}_A$ and $l \in \mathcal{V}_B$. Assume $d_i, d_k, d_j, d_l \geq 2$. Then*

$$\Gamma_C(p, q) \geq \psi(i, j, k, l)\Gamma_A(i, j)\Gamma_B(k, l),$$

where

$$\psi(i, j, k, l) = \frac{(d_i - 1)(d_k - 1)(d_j - 1)(d_l - 1)}{(d_i d_k - 1)(d_j d_l - 1)}.$$

Note that $\psi(i, j, k, l) \in \left[\frac{1}{9}, 1\right)$.

Proof. Thm. 5 shows $\Diamond_{pq} = \Diamond_{ij}\Diamond_{kl} + \eta$, where $\eta > 0$ for $d_i, d_k, d_j, d_l \geq 2$. Recall $d_p = d_i d_k$ and $d_q = d_j d_l$. Thus,

$$\begin{aligned}\Gamma_C(p, q) &= \frac{\Diamond_{pq}}{(d_p - 1)(d_q - 1)} \\ &> \frac{\Diamond_{ij}\Diamond_{kl}}{(d_i d_k - 1)(d_j d_l - 1)} \\ &= \psi(i, j, k, l)\Gamma_A(i, j)\Gamma_B(k, l).\end{aligned}$$

□

Note that we opted for simplicity over the tightest bounds possible. Typically, $\Diamond_{pq}$ is much greater than $\Diamond_{ij}\Diamond_{kl}$, and the ratio in the scaling law is better than $\psi(i, j, k, l)$.

C. Community Structure

We take the definition of bipartite community structure within a bipartite graph $\mathcal{G}_A$ to be a connected subset of $\mathcal{U}_A \cup \mathcal{W}_A$ with relatively high internal edge density and relatively low external edge density. In this section, we demonstrate that if bipartite factors A and B have strong community structure, then this feature is maintained by the Kronecker graph that has all self loops added to factor A.

Def. 11. (Internal/External Counts and Densities) *Let $\mathcal{S}_A \subset \mathcal{V}_A$ and let $\mathcal{R}_A \subset \mathcal{U}_A$, $\mathcal{T}_A \subset \mathcal{W}_A$ such that $\mathcal{S}_A = \mathcal{R}_A \cup \mathcal{T}_A$ and $\mathcal{R}_A \cap \mathcal{T}_A = \emptyset$. Define* indicator vector $\mathbf{1}_{\mathcal{S}_A} \in \mathbb{B}^{n_A}$ *to have a 1 in the i-th entry if and only if $i \in \mathcal{S}_A$. The* internal edge count *is*

$$m_{in}(\mathcal{S}_A) := \frac{1}{2}\mathbf{1}_{\mathcal{S}_A}^t A \mathbf{1}_{\mathcal{S}_A},$$

whereas the external edge count *is*

$$m_{out}(\mathcal{S}_A) = \mathbf{1}_{\mathcal{S}_A}^t A \left(\mathbf{1}_A - \mathbf{1}_{\mathcal{S}_A}\right).$$

Internal and external edge densities *are*

$$\rho_{in}(\mathcal{S}_A) := \frac{m_{in}(\mathcal{S}_A)}{|\mathcal{R}_A||\mathcal{T}_A|},$$

and

$$\rho_{out}(\mathcal{S}_A) := \frac{m_{out}(\mathcal{S}_A)}{|\mathcal{R}_A||\mathcal{W}_A| + |\mathcal{U}_A||\mathcal{T}_A| - 2|\mathcal{R}_A||\mathcal{T}_A|}$$

Def. 12. (Product of Sets) *Let A, B be bipartite, undirected, and connected, and let $C = (A + I_A) \otimes B$. The* Kronecker product of connected vertex sets $\mathcal{S}_A \in \mathcal{V}_A$ and $\mathcal{S}_B \in \mathcal{V}_B$ *is*

$$\mathcal{S}_C := \mathcal{S}_A \otimes \mathcal{S}_B := \mathit{supp}(\mathbf{1}_A \otimes \mathbf{1}_B).$$

Then let $\mathcal{R}_A = \mathcal{S}_A \cap \mathcal{U}_A$ and $\mathcal{T}_A = \mathcal{S}_A \cap \mathcal{W}_A$. Then define

$$\mathcal{R}_C := \{\mathcal{R}_A \otimes \mathcal{R}_B\} \cup \{\mathcal{T}_A \otimes \mathcal{R}_B\},$$

and

$$\mathcal{T}_C := \{\mathcal{R}_A \otimes \mathcal{T}_B\} \cup \{\mathcal{T}_A \otimes \mathcal{T}_B\}.$$

Note that $|\mathcal{R}_C| = |\mathcal{S}_A||\mathcal{R}_B|$ and $|\mathcal{T}_C| = |\mathcal{S}_A||\mathcal{T}_B|$.

Thm. 7. (Internal/External Edge Counts) *Let A, B be bipartite, undirected, and connected, and let $C = (A+I_A)\otimes B$. Then,*

$$m_{in}(\mathcal{S}_C) = 2m_{in}(\mathcal{S}_A)m_{in}(\mathcal{S}_B) + |\mathcal{S}_A|m_{in}(\mathcal{S}_B),$$

and

$$\begin{aligned} m_{out}(\mathcal{S}_C) &= m_{out}(\mathcal{S}_A)m_{out}(\mathcal{S}_B) + 2m_{out}(\mathcal{S}_A)m_{in}(\mathcal{S}_B) \\ &\quad + |\mathcal{S}_A|m_{out}(\mathcal{S}_B) + 2m_{in}(\mathcal{S}_A)m_{out}(\mathcal{S}_B). \end{aligned}$$

We use Thm. 7 to show that $\mathcal{G}_C$ has sets whose internal density follows a controlled scaling law (bounded from below), as long as the ratio of parts of $\mathcal{S}_A = \mathcal{R}_A \cup \mathcal{T}_A$ is modest.

Cor. 1. *Assume*

$$\omega := \min(|\mathcal{R}_A|/|\mathcal{S}_A|, |\mathcal{T}_A|/|\mathcal{S}_A|),$$

and note $\omega \in [|\mathcal{S}_A|^{-1}, 1/2]$. We have

$$\rho_{in}(\mathcal{S}_C) \geq 2\omega\rho_{in}(\mathcal{S}_A)\rho_{in}(\mathcal{S}_B).$$

Proof.

$$\rho_{in}(\mathcal{S}_C) = \frac{2m_{in}(\mathcal{S}_C)}{|\mathcal{R}_C||\mathcal{T}_C|} = \frac{2(2m_{in}(\mathcal{S}_A)m_{in}(\mathcal{S}_B)+|\mathcal{S}_A|m_{in}(\mathcal{S}_B))}{|\mathcal{S}_A|^2|\mathcal{R}_B||\mathcal{T}_B|} > \frac{2(2m_{in}(\mathcal{S}_A)m_{in}(\mathcal{S}_B))}{|\mathcal{S}_A|^2|\mathcal{R}_B||\mathcal{T}_B|} = 4\theta_{|\mathcal{S}_A|,|\mathcal{R}_A|,|\mathcal{T}_A|} \cdot \rho_{in}(\mathcal{S}_A)\rho_{in}(\mathcal{S}_B),$$

where

$$\theta_{|\mathcal{S}_A|,|\mathcal{R}_A|,|\mathcal{T}_A|} := \frac{|\mathcal{R}_A||\mathcal{T}_A|}{|\mathcal{S}_A|^2}.$$

which is greater than $\omega/2$, as implied by (assuming, without loss of generality, that $|\mathcal{R}_A| \leq |\mathcal{T}_A|$),

$$\frac{|\mathcal{R}_A||\mathcal{T}_A|}{|\mathcal{S}_A|^2} = \frac{|\mathcal{R}_A|}{|\mathcal{S}_A|}\left(1 - \frac{|\mathcal{R}_A|}{|\mathcal{S}_A|}\right) = \omega(1-\omega) \geq \frac{\omega}{2}.$$

□

Cor. 2. *Define*

$$\epsilon_{\mathcal{S}_A,\mathcal{S}_B} := \max\left(\frac{|\mathcal{S}_A|}{|\mathcal{V}_A|}, \frac{|\mathcal{R}_B|}{|\mathcal{U}_B|}, \frac{|\mathcal{T}_B|}{|\mathcal{W}_B|}\right),$$

$$\xi_{\mathcal{S}_A} := \frac{2m_{in}(\mathcal{S}_A) + |\mathcal{S}_A|}{m_{out}(\mathcal{S}_A)},\ \xi_{\mathcal{S}_B} := \frac{2m_{in}(\mathcal{S}_B) + |\mathcal{S}_B|}{m_{out}(\mathcal{S}_B)}.$$

We have

$$\rho_{out}(\mathcal{S}_C) \leq \frac{(1+\xi_{\mathcal{S}_A})(1+\xi_{\mathcal{S}_B})}{1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}}\rho_{out}(\mathcal{S}_A)\rho_{out}(\mathcal{S}_B).$$

Proof. First, we show a relationship for the numerator of $\rho_{min}(\mathcal{S}_C)$ and the numerators of $\rho_{min}(\mathcal{S}_A)$ and $\rho_{min}(\mathcal{S}_A)$.

$$\begin{aligned} m_{out}(\mathcal{S}_C) &= m_{out}(\mathcal{S}_A)m_{out}(\mathcal{S}_B) + \\ &\quad 2m_{out}(\mathcal{S}_A)m_{in}(\mathcal{S}_B) + |\mathcal{S}_A|m_{out}(\mathcal{S}_B) \\ &\quad 2m_{in}(\mathcal{S}_A)m_{out}(\mathcal{S}_B) \\ &< (m_{out}(\mathcal{S}_A) + 2m_{in}(\mathcal{S}_A) + |\mathcal{S}_A|) \cdot \\ &\quad (m_{out}(\mathcal{S}_B) + 2m_{in}(\mathcal{S}_B) + |\mathcal{S}_B|) \\ &\leq (1+\xi_{\mathcal{S}_A})(1+\xi_{\mathcal{S}_B})m_{out}(\mathcal{S}_A)m_{out}(\mathcal{S}_B) \end{aligned}$$

Now we use the definition of $\epsilon_{\mathcal{S}_A,\mathcal{S}_B}$ to show a relation for the denominator of $\rho_{min}(\mathcal{S}_C)$, $denom(\mathcal{S}_C) :=$

$$\begin{aligned} &= |\mathcal{R}_C||\mathcal{W}_C| + |\mathcal{U}_C||\mathcal{T}_C| - 2|\mathcal{R}_C||\mathcal{T}_C| \\ &= |\mathcal{S}_A||\mathcal{V}_A|\left(|\mathcal{R}_B||\mathcal{W}_B| + |\mathcal{U}_B||\mathcal{T}_B|\right) \\ &\quad -2|\mathcal{S}_A|^2|\mathcal{R}_B||\mathcal{T}_B| \\ &> \left(1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}\right)|\mathcal{S}_A||\mathcal{V}_A|\left(|\mathcal{R}_B||\mathcal{W}_B| + |\mathcal{U}_B||\mathcal{T}_B|\right) \\ &= \left(1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}\right)\left(|\mathcal{R}_A| + |\mathcal{T}_A|\right)\left(|\mathcal{U}_A| + |\mathcal{W}_A|\right) \cdot \\ &\quad \left(|\mathcal{R}_B||\mathcal{W}_B| + |\mathcal{U}_B||\mathcal{T}_B|\right) \\ &> \left(1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}\right)\left(|\mathcal{R}_A||\mathcal{W}_A| + |\mathcal{U}_A||\mathcal{T}_A|\right) \cdot \\ &\quad \left(|\mathcal{R}_B|\mathcal{W}_B| + |\mathcal{U}_B||\mathcal{T}_B|\right) \\ &> \left(1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}\right)\left(|\mathcal{R}_A||\mathcal{W}_A| + |\mathcal{U}_A||\mathcal{T}_A| - 2|\mathcal{R}_A||\mathcal{T}_A|\right) \cdot \\ &\quad \left(|\mathcal{R}_B|\mathcal{W}_B| + |\mathcal{U}_B||\mathcal{T}_B| - 2|\mathcal{R}_B||\mathcal{T}_B|\right) \\ &= \left(1-\epsilon^2_{\mathcal{S}_A,\mathcal{S}_B}\right) denom(\mathcal{S}_A)denom(\mathcal{S}_B) \end{aligned}$$

Combining this with the relation for the numerator gives the result. □

IV. Experiments

We downloaded a small bipartite graph dataset to validate our techniques, the unicode language network (`unicode`) from Konect [6]. This is a small, disconnected adjacency matrix A with hundreds of vertices, over a thousand edges, and 1662 global 4-cycles. We formed $C = (A+I_A)\otimes A$, yielding a bipartite graph with hundreds of thousands of vertices, millions of edges, and 946,565,889 global 4-cycles. We summarize the statistics of the factor and Kronecker product in Tab. I.

The Kronecker structure of $\mathcal{G}_C$ allows functions of several powers of C to be computed relatively cheaply, for example

$$diag(C^4) = diag(A^4 + 6A^2 + I_A) \otimes diag(A^4).$$

This means that the formulas in § III can be applied to compute local and global square counts of $\mathcal{G}_C$ with $\mathcal{O}(|\mathcal{V}_A|^{2.3729})$ complexity with dense matrix vector products, which is $\mathcal{O}(|\mathcal{V}_C|^{1.1865})$. This is a nearly linear complexity in $|\mathcal{E}_C|$ and is fairly palatable for fairly large graphs. For this example, the local and global 4-cycle counts are done on seconds on a commodity laptop. The vertex degree versus the vertex participation in 4-cycles is plotted on a log-log scale in Fig 5.

Adjacency	Vertices	Edges	Global 4-Cycles
A	$\|\mathcal{U}_A\| = 254, \|\mathcal{W}_A\| = 614$	$\|\mathcal{E}_A\| = 1,256$	1,662
$C = (A + I_A) \otimes A$	$\|\mathcal{U}_C\| = 220,472, \|\mathcal{W}_C\| = 532,952$	$\|\mathcal{E}_C\| = 3,155,072$	946,565,889

TABLE I: Graph statistics for `unicode` and a Kronecker product graph built using `unicode` as both factors with all self loops in one factor.

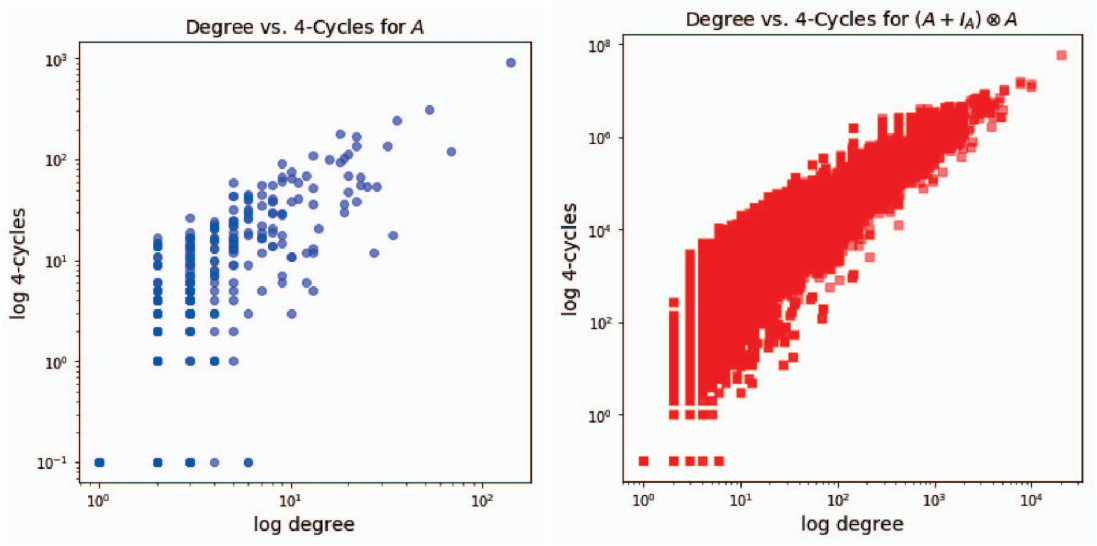


Fig. 5: Vertex degree versus count of 4-cycles for `unicode` and a Kronecker product graph built using `unicode` as both factors with all self loops in one factor. Zero values are mapped to 10^{-1}.

V. Conclusion

We described a few approaches to generating bipartite Kroncker product graphs that are connected. For these approaches, we derived Kronecker formulas that provide ground truth for several bipartite graph properties, including local 4-cycle counts at vertices and edges. We also demonstrate scaling laws that show that clustering coefficients and community structure are both bounded and controllable. This implies relatively dense structures in the factors yield relatively dense structures in the product. The ground truth values are computable in linear time, making this approach extremely attractive for generating massive bipartite graphs with ground truth and similar challenges to real-world massive graphs.

For future work, we intend to use nonstochastic Kronecker graphs to benchmark and validate massive-scale bipartite graph pattern matching algorithms that include 4-cycle counting. Also, we intend to implement this style of generator in a distributed version of graphBLAS, including using the ground truth formulas derived here to compute ground truth values during generation.

Appendix

A. Properties of Kronecker and Hadamard Products

Prop. 1. (Properties of Kronecker Product [1], [21], [22])

(a) Scalar Multiplication. *For any* $a_1, a_2 \in \mathbb{R}$,

$$(a_1 a_2)(A_1 \otimes A_2) = (a_1 A_1) \otimes (a_2 A_2).$$

(b) Distributivity.

$$\begin{aligned}(A_1 + A_2) \otimes A_3 &= (A_1 \otimes A_3) + (A_2 \otimes A_3) \quad \textit{and}\\ A_1 \otimes (A_2 + A_3) &= (A_1 \otimes A_2) + (A_1 \otimes A_3).\end{aligned}$$

(c) Transposition. $(A_1 \otimes A_2)^t = (A_1^t \otimes A_2^t)$.

(d) Matrix-Matrix Multiplication. *When* $n_{A_1} = m_{A_3}$ *and* $n_{A_2} = m_{A_4}$,

$$(A_1 \otimes A_2)(A_3 \otimes A_4) = (A_1 A_3) \otimes (A_2 A_4).$$

Prop. 2. (Properties of Hadamard Product [22]) *In the following, we implicitly assume that* $n_{A_0} = n_{A_1}$ *and* $m_{A_0} = m_{A_1}$ *whenever* $A_0 \circ A_1$ *is present.*

(a) Commutativity. $A_1 \circ A_2 = A_2 \circ A_1$.

(b) Scalar Multiplication. *For any* $a_1, a_2 \in \mathbb{R}$,

$$(a_1 a_2)(A_1 \circ A_2) = (a_1 A_1) \circ (a_2 A_2).$$

(c) Distributivity.

$$\begin{aligned}(A_1 + A_2) \circ A_3 &= (A_1 \circ A_3) + (A_2 \circ A_3) \quad \textit{and}\\ A_1 \circ (A_2 + A_3) &= (A_1 \circ A_2) + (A_1 \circ A_3).\end{aligned}$$

(d) Transposition. $(A_1 \circ A_2)^t = (A_1^t \circ A_2^t)$.

(e) Hadamard-Kronecker Distributivity.

$$(A_1 \otimes A_2) \circ (A_3 \otimes A_4) = (A_1 \circ A_3) \otimes (A_2 \circ A_4).$$

(f) Diagonal-Kronecker Distributivity. *When* $m_{A_1} = n_{A_1}$ *and* $m_{A_2} = n_{A_2}$,

$$\textit{diag}(A_1 \otimes A_2) = \textit{diag}(A_1) \otimes \textit{diag}(A_2).$$

REFERENCES

[1] P. M. Weichsel, "The Kronecker product of graphs," *Proceedings of the American Mathematical Society*, vol. 13, no. 1, pp. 47–52, 1962.

[2] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, "Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2005, pp. 133–145.

[3] T. Steil, B. W. Priest, G. Sanders, R. Pearce, T. L. Fond, and K. Iwabuchi, "Distributed kronecker graph generation with ground truth of many graph properties," in *IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2019, Rio de Janeiro, Brazil, May 20-24, 2019*, 2019, pp. 251–260. [Online]. Available: https://doi.org/10.1109/IPDPSW.2019.00048

[4] A. E. Sarıyüce and A. Pinar, "Peeling bipartite networks for dense subgraph discovery," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2018, p. 504512. [Online]. Available: https://doi.org/10.1145/3159652.3159678

[5] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, jun 2014.

[6] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, 2013, pp. 1343–1350. [Online]. Available: http://userpages.uni-koblenz.de/ kunegis/paper/kunegis-koblenz-network-collection.pdf

[7] T. Davis and Y. Hu, "The university of florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, p. 1, 11 2011.

[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, 1998.

[9] J. Cohen, "Trusses: Cohesive subgraphs for social network analysis," *web*. [Online]. Available: http://www2.computer.org/cms/Computer.org/dl/mags/cs/2009/04/extras/msp2009040029s1.pdf

[10] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips, "Tolerating the community detection resolution limit with edge weighting." *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 83 5 Pt 2, p. 056119, 2011.

[11] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016. [Online]. Available: http://science.sciencemag.org/content/353/6295/163

[12] G. Sanders, R. Pearce, T. L. Fond, and J. Kepner, "On large-scale graph generation with validation of diverse triangle statistics at edges and vertices," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2018, pp. 287–296.

[13] R. Pearce, T. Steil, B. W. Priest, and G. Sanders, "One quadrillion triangles queried on one million processors," in *2019 IEEE High Performance Extreme Computing Conference, HPEC 2019, Waltham, MA, USA, September 24-26, 2019*, 2019, pp. 1–5. [Online]. Available: https://doi.org/10.1109/HPEC.2019.8916243

[14] G. Robins and M. Alexander, "Small worlds among interlocking directors: Network structure and distance in bipartite graphs," *Computational & Mathematical Organization Theory*, vol. 10, no. 1, pp. 69–94, May 2004. [Online]. Available: https://doi.org/10.1023/B:CMOT.0000032580.12184.c0

[15] P. Zhang, J. Wang, X. Li, Z. Di, and Y. Fan, "Clustering coefficient and community structure of bipartite networks," 2008.

[16] T. Opsahl, "Triadic closure in two-mode networks: Redefining the global and local clustering coefficients," *arXiv e-prints*, p. arXiv:1006.0887, Jun 2010.

[17] Z. Zou, "Bitruss decomposition of bipartite graphs," in *Proceedings, Part II, of the 21st International Conference on Database Systems for Advanced Applications - Volume 9643*, ser. DASFAA 2016. Berlin, Heidelberg: Springer-Verlag, 2016, p. 218233. [Online]. Available: https://doi.org/10.1007/978-3-319-32049-6_14

[18] N. Alon, R. Yuster, and U. Zwick, "Finding and counting given length cycles," *Algorithmica*, vol. 17, no. 3, pp. 209–223, Mar 1997. [Online]. Available: https://doi.org/10.1007/BF02523189

[19] J. Cohen, "Graph twiddling in a mapreduce world," *Computing in Science and Engg.*, vol. 11, no. 4, pp. 29–41, Jul. 2009.

[20] J. Kepner, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, T. Davis, V. Gadepally, M. Houle, M. Hubbell, H. Jananthan, M. Jones, A. Klein, P. Michaleas, R. Pearce, L. Milechin, J. Mullen, A. Prout, A. Rosa, G. Sanders, C. Yee, and A. Reuther, "Design, generation, and validation of extreme scale power-law graphs," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2018, pp. 279–286.

[21] C. F. Van Loan, "The ubiquitous Kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 85–100, 2000.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY, USA: Cambridge University Press, 2012.

[23] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A recursive model for graph mining," in *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 2004, pp. 442–446.

[24] J. Ang, B. Barrett, K. Wheeler, and R. Murphy, "Introducing the graph 500," 01 2010. [Online]. Available: https://graph500.org/

[25] S. Samsi, V. Gadepally, M. Hurley, M. Jones, E. Kao, S. Mohindra, P. Monticciolo, A. Reuther, S. Smith, W. Song, D. Staheli, and J. Kepner, "Static graph challenge: Subgraph isomorphism," in *High Performance Extreme Computing Conference (HPEC)*. IEEE, 2017.

[26] E. Kao, V. Gadepally, M. Hurley, M. Jones, J. Kepner, S. Mohindra, P. Monticciolo, A. Reuther, S. Samsi, W. Song, D. Staheli, and S. Smith, "Streaming Graph Challenge - Stochastic Block Partition," in *High Performance Extreme Computing Conference (HPEC)*. IEEE, 2017.

[27] S. Aksoy, T. G. Kolda, and A. Pinar, "Measuring and Modeling Bipartite Graphs with Community Structure," *arXiv e-prints*, p. arXiv:1607.08673, Jul 2016.

[28] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, mar 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756039

[29] J. Kepner and J. Gilbert, *Graph algorithms in the language of linear algebra*. SIAM, 2011.

[30] J. Kepner, P. Aaltonen, D. Bader, A. Buluç, F. Franchetti, J. Gilbert, D. Hutchison, M. Kumar, A. Lumsdaine, H. Meyerhenke, S. McMillan, J. Moreira, J. Owens, C. Yang, M. Zalewski, and T. Mattson, "Mathematical foundations of the GraphBLAS," in *IEEE High Performance Extreme Computing (HPEC)*, 2016.

[31] A. Buluç, T. Mattson, S. McMillan, J. Moreira, and C. Yang, "The GraphBLAS C API Specification, version 1.0.0," May 2017, http://graphblas.org.

[32] ——, "The GraphBLAS C API Specification, version 1.3.0," The GraphBLAS Signatures Subgroup, Tech. Rep., September 2019, http://graphblas.org.