Journal of Systemics, Cybernetics and Informatics (2022) 20(2), 23-32
https://doi.org/10.54808/JSCI.20.02.23

Graph Sampling Through Graph Decomposition and Reconstruction Based on Kronecker Graphs

Shen LU
Department of Computer Science & Engineering, University of South Florida
Tampa, Florida, 33620, USA

Les PIEGL
Department of Computer Science & Engineering, University of South Florida
Tampa, Florida, 33620, USA

Richard S. SEGALL
Department of Information Systems & Business Analytics, Arkansas State University
State University, Arkansas, 72467, US

ABSTRACT[1]

The connectedness of the social network gives rise to a new challenge of how to efficiently sample the network and keep the graph properties and topology properties as well. Inspired by R-MAT and Kronecker graph generators, based on the observation of different graph topology types, we proposed to use Kronecker graph as the prime graph to conduct Kronecker graph double cover through periphery subgraphs. First of all, the connectedness of the graph remains during graph merging. Secondly, only redundant vertices and edges are merged so that the characteristics of the graphs are kept. Also, graph merging only works on periphery subgraphs from low degrees to higher up so those topology properties are kept. Finally, although some edges are merged, since the similarity groups generated based on Kronecker graph similarity is independent of the degree distribution, Kronecker double cover operation does not affect the graph degree centrality measure. We theoretically prove the feasibility of the Kronecker double-cover operation and also compare the quality of the sample set with Snowball sampling and Es-i sampling sets. Experimental results show us that, when the separation of the core and periphery subgraphs is between mean (the average of the degrees) and mean+std (standard deviation of the degrees), the topology types and graph properties can be preserved. This conclusion confirms the existence of the topology types, and also proves the topology types of the real-world graphs are not random.

Keywords: Graph sampling, Topology types, Cartesian Products, Kronecker Double Cover, Graph Density, Graphon

1. INTRODUCTION

Based on the self-similar nature, graphs can be generated through edges, as shown in [6], and subgraphs, as shown in [13]. We want to apply these ideas to graph sampling that, given a graph, instead of dropping subgraphs to data sets we define a recursive process to consistently merge similar subgraphs to shrink sample set sizes.

The subgraphs are selected based on the prime graph, such as R-MAT and Kronecker graphs. In comparison with graph generators, the merging process is based on original graphs so that the impact on the graph properties and graph topology types can be evaluated during graph sampling.

Network research has been conducted for many years, especially in the fields of physics and mathematics. There are many different types of networks, such as social networks, power grid networks, communication networks, and so on. The structure of the network is represented by graphs in the data structure, and by an adjacent matrix in mathematics. For finite graphs and graph morphisms, the connectedness can be preserved during projection and join operations and be recovered under pullbacks [8]. This property can ensure that graph merge and graph product can preserve the connectedness of the graph. In comparison with random sampling, it is much easier to maintain the graph properties and topology properties through graph merging.

In this paper, we discuss related work in section 2, section 3 introduces related theory for interesting layer construction and network decomposition. In section 4, we discuss the construction of interesting layers. In section 5, we present experiments on several data sets. We conclude our work in section 6.

[1] Acknowledgement is made to Venkata Jaipal Reddy, Batthula, of University of Arkansas at Little Rock (UALR) for editorial review of final submission.

2. RELATED WORK

Graph Theories and Operations

For finite graphs without loops or multiple edges, the operations we need for graph sampling include graph decomposition, reconstructible graph, X-graph join, and Kronecker double covers. These operations have been generally discussed.

The conditions of indecomposable and decomposable graphs were given in [15] and it also indicated that conversion from indecomposable graphs to decomposable graphs can be done by removing some edges.

Graph automorphisms, connectedness and partition of joined graphs, especially X-join graphs, were defined in [12]. Graph X-join operation is to replace each x of X by graph Y_x. The problem of finding necessary conditions that G(X*Y) consists precisely of those automorphisms induced by automorphisms of G(X) can be used to determine the topology of X-join graphs. When we apply graph X-join to symmetric graphs, such as Kronecker graphs, the connectedness and the properties of the topology of jointed graphs can be verified.

Graph double cover is a graph projection operation given in [16]. Similar to Cartesian product of two graphs, graph double cover checks both of the two vertices on each edge but merge both vertices if they are similar, which is a two-fold projection onto G and preserves local structure as well. For graph sampling, to keep the local structure, we choose to use a graph double cover to fold similar edges and vertices instead of removing them.

The local projection of graphs was discussed in [8]. When graphs can be divided into subsets, local projection can be conducted through graph product, n-fold cover, and other graph operations. Functional operations can be defined in categorical graphs. These operations map vertices to either the same vertices, in which vertices are merged or different vertices, in which new edges are built.

When conducting graph operations, graph properties can be estimated with Additive combinatorics and extreme graph theory [17]. For example, the maximum number of edges, the maximum number of distinct distances, the maximum number of triangles, and other quantities of the graphs can be estimated during edge removal, subgraph removal, randomized construction, and algebraic construction.

Graph Generators

Graphs can be generated recursively with individual graph patterns, such as R-MAT [6], and Kronecker Graphs [13]. This gives rise to the question if we can sample the graph by recursively factorizing the graph with these meaningful graph patterns and finding the useful portion of the graph, meanwhile, graph topology and graph properties can be maintained.

R-MAT [6] partition a social network into four regions: partition a and d represent separate groups of nodes that correspond to communities, and partition b and c are the cross-links between these two groups, such as friends of separate interests. Although R-MAT algorithm locates edges into an adjacent matrix, the distribution of edges in the four regions and the correlation among edges in the four regions are random and not related to topology types. Given the definition of the four regions, we have to define which edges belong to the core set and which edges belong to the periphery set, which cannot be done by randomly dropping edges into different regions.

In [1], the combing problem in R-MAT generator was discussed that the graph distribution is combed at regular geometric intervals. The symptom of this combing problem is that certain degrees are not showing and those are between regular intervals. The combing problems confirm the weakness of R-MAT generator that, other than the meanings of the edges in four regions, the topology types of the graphs also need to be simulated.

Similar to R-MAT, in [1], sample sets are constructed by randomly choosing edges. The methodology selected edges randomly with no regard to the four regions and the probability of edges in the four regions. The idea is that randomly selected edges can be used as a seed set so that the sample set can be expanded by adding the connections between vertices in a sample set. In practice, after the seed set of edges is selected, only a few connections can be found between those vertices in the seed set.

Kronecker product [13] takes advantage of the symmetric structure which can ensure to produce of self-similar graphs of any size, match the combination of graph properties, and simplify graph operations to iterative processes, such as graph join and projection. Kronecker graph structure is also the smallest symmetric graph pattern which can be used as a prime graph for the graph computation. Kronecker product can also be used to factorize and expand graphs [4] [5].

Graph Sampling

Snowball sampling [9] is a random vertex sampling methodology. The sampling process starts with k individual vertices, each individual vertex call k more individuals, this process is conducted repeatedly in s stages. For networked data, multiple-stage sampling is a better option, so that the difference between the sample sets and the original data can be observed.

In [14], graphs are divided into low-dimensional embeddings of the neighborhood information of each node. The number of vertices and edges are not changed but the graphs are chopped into small chunks. The neighborhood information carries both the local and the global properties of the graph and maintains the topology types as well.

3. PROBLEM DEFINITIONS

Notation: The following symbols are used throughout the paper. Let V(X) be the vertex set of the graph X, and E(X) be the edge set of the graph X. An edge {v,w} is often denoted as vw. $v \sim_X w$ denotes the adjacency of the vertices v and w in graph X.

Topology Types

The topology type of the network represents the interaction between vertices and the connectedness of the vertices. Network topology carries significant properties of the network. Once the network type is defined, both the topology type and the topology properties need to be kept for different network operations. In graph sampling, topology properties can be used to measure if the data set and the sample set are the same or different. Six different topology types are summarized in figure 1 and indicate the difference in randomness and regularity of the vertices and connections.

From ring lattice network to small-world network and Erdos random network, the connections between vertices become more random. From Core-periphery networks to scale-free networks and cellular networks, the connections are more decentralized and more distributed into subgroups. When graph sampling is applied, a sampling strategy needs to be designed based on different topology types.

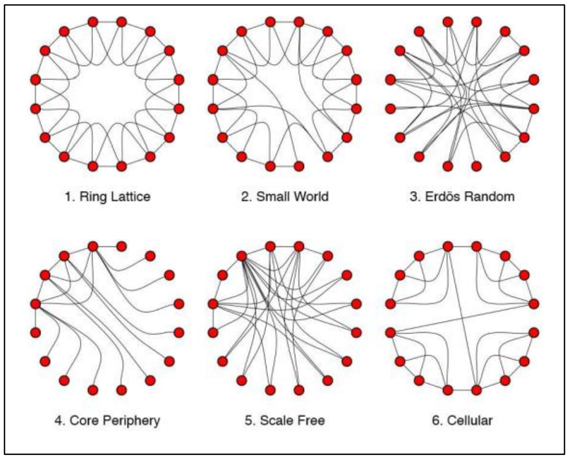


Fig. 1. Pure Topology Types [Albert, R. et.al. 2002]

Automorphism

Automorphism: Given a graph x, a permutation α of V(X) is an automorphism of X if for all $\mu, v \ \in V(X)$

$$\{\mu, v\} \in E(X) \Leftrightarrow \{\alpha(\mu), \alpha(v)\} \in E(X) \qquad (1)$$

The set of all automorphisms of a graph X, under the operation of composition of functions, forms a subgroup of the symmetric group on V(X) called the ***automorphism group*** of X, and it is denoted Aut(X).

From the definition of graph automorphism, we can derive these facts in automorphism groups, that, let the components of X be X_1, …, X_k then

$$Aut(X) = \prod_{i=1}^{k} Aut(X_i) \qquad (2)$$

Also, for a simple graph X with edge-complement $\overline{X}$, we have

$$Aut(X) = Aut(\overline{X}) \qquad (3)$$

Transitivity: A group X of perms of a set S acts transitively or is transitive on S if, for every x, y ∈ S, there exists $\alpha \in X$ such that $\alpha(x) = y$ is vertex-transitive, Aut(X) acts transitively on V(X), and acts doubly transitively on S if, for any two ordered pairs of distinct elements (x_1, x_2), (y_1, y_2) ∈ S*S, there exists $\alpha \in G$ such that $\alpha(x_1) = y_1$ and $\alpha(x_2) = y_2$

Graphon: A **graphon** (graph function) is symmetric measurable function W: $[0,1]^2$ -> [0,1]

Graph Homomorphism: A **graph homomorphism** from H to G is a map $\varphi: V(H) \to V(G)$ such that if $\mu v \in E(H)$.

Graph Homomorphism Density: Let $\hom(H, G)$ be the set of all such homomorphisms and let $\hom(H, G) = |\hom\ (H, G)|$. Define **homomorphism density** as

$$t(H, G) = \frac{\hom\ (H,G)}{|V(G)|^{|V(H)|}} \qquad (4)$$

This is also the probability that a uniformly random map is a homomorphism.

Edge Density: Let X and Y be sets of vertices in a graph G. Let e_G(X,Y) be the number of edges between X and Y; that is

$$e_G(X, Y) = |\{(X, Y) \in (X \times Y | xy \in E(G)\}| \qquad (5)$$

From this, we can define the **edge density** between X and Y to be

$$d_G(X, Y) = \frac{e_G(X,Y)}{|X||Y|} \qquad (6)$$

Deck: Given a graph X, the collection of its vertex deleted subgraphs X-v for all v∈V(X) is called the deck of X and is denoted by D(X).

Convergent: The sequence converges to W if t(H, X_n) (or t(H, W_n)) converges to t(H, W) for every graph H.

Existence of Limit: Every convergent sequence of graphs or graphons has a limit graphon.

Equivalence of Convergence: A sequence of graphs or graphons is convergent if and only if it is a Cauchy sequence with respect to the cut (distance) metric. (A Cauchy sequence with respect to metric d is a sequence {x_i} that satisfies $sup_{m \geq 0} d(x_n, x_{n+m}) \to 0\ as\ n \to \infty$)

Kronecker Double Cover: Kronecker double cover $\tilde{X}$ of a graph X has vertices $(v$,1) and $(v$,2) for each vertex v of X, with adjacency $v \sim_x$ w, if and only if $(v, 1) \sim_x (w, 2)$ and $(v, 2) \sim_x (w, 1)$ in $\tilde{X}$

On vertex

$$\left.\begin{matrix}(v,1)\\(v,2)\end{matrix}\right\} \Longrightarrow v \qquad (7)$$

On edges

$$\left.\begin{matrix}(v,1)(w,2)\\(v,2)(w,1)\end{matrix}\right\} \Longrightarrow (v,w) \tag{8}$$

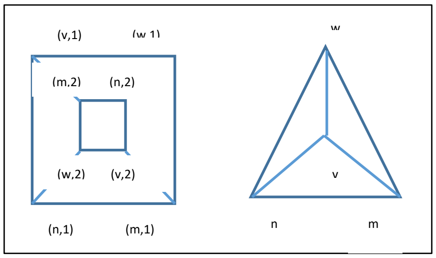


Fig. 2. Kronecker Double Cover

Graph Decomposition and Reconstruction

Graphs are self-similar. Based on the characteristics of graph structure, we want to use a prime graph as the basic unit to extract similar patterns. The prime graph is the combination of vertices and edges that can carry more information and be specific to a particular application.

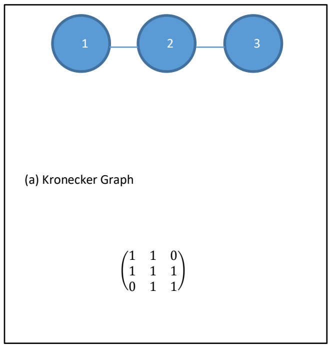


Fig. 3. Kronecker Graph

The prime graph we use to factorize the network is called Kronecker graph, as shown in figure 3. The reason we choose Kronecker graph as the prime graph is because it is the smallest symmetric graph without loops. The symmetric property of the Kronecker graph can ensure that graphs can expand and shrink to any size through Kronecker operations.

Graph decomposition and reconstruction require that the graphs are reconstructible, the homomorphism density can be convergent as n goes to infinity and the upper and lower bounds of the maximum number of copies of the prime graphs can be determined. The following theorems provide restrictions to satisfy these criteria.

Theorem 1. Every graph with at least three vertices is reconstructible [11].

Theorem 2. Every graph on at least four edges is edge-reconstructible [11].

Theorem 1 and Theorem 2 show us the lower bounds of the vertices and the edges for reconstructible graphs. We also care about the upper bounds of the vertices and the edges so that, after factorization, the size of the subgraphs can be loaded into the main memory. We prove the upper bounds in lemma 4.

Other than the upper bounds and lower bounds of the vertices and edges, we also want to know the number of possible subgraphs, after factorization, so that we can ensure the copies of the subgraphs can be loaded into memory space.

Theorem 3. Given a graph G-v in the deck of G, the degree of v and the degrees of the neighbors of v in G are reconstructible [11].

Theorem 4. Suppose G and F are graphs with |V(F)|<|V(G)|. Then $(|V(G)|-|V(F)|)\binom{G}{F} = \sum_{v\in V(G)}\binom{G-v}{F}$, therefore, $\binom{G}{F}$ is reconstructible. [11]

The total copies of the subgraphs of G choosing F can be quantitatively computed. After quantitatively evaluating the upper bounds and the lower bounds of the space complexity, we want to show how to reconstruct subgraphs. The **deck** of G, denoted by D(G), is formed by removing vertices from the graph G. When a vertex is removed from the graph, connected edges are also removed, if any. D(G) contains subgraphs of G. When we factorize the graph with prime graphs, the results are the subset of the deck.

Theorem 5. Let G be a graph without isolated vertices. The deck of G is edge-reconstructible, that is D(G) is uniquely determined from $\varepsilon D(G)$. Therefore, if G is reconstructible, then it is also edge-reconstructible [11].

Based on theorem 5, we know graph factorization with prime graphs are conductible because the deck of G is edge-reconstructible so that the subset of the deck of G is reconstructible which is the factorization results we are looking for.

Theorem 6. Let A be a n*n symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and let B be obtained from A by removing its ith row and ith column and suppose B has eigenvalues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$, then the eigenvalues of B interlace those of A, that is, $\lambda_i \geq \mu \geq \lambda_{i+1}$ [10]

Based on theorem 6, after reconstruction, the order of the eigenvalues remains.

Given Kronecker graph G$_1$, as shown in figure 4(a), as n goes to infinity, the graphon W_{Gn}: $[0,1]^2 \rightarrow [0,1]$ converges to function fig 4(c) as graph properties remain.

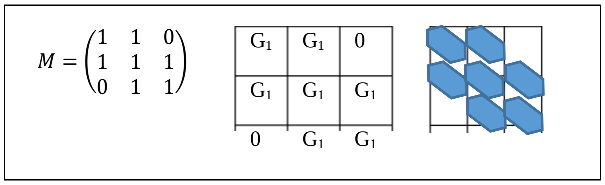


Fig. 4. Kronecker Graph G$_1$ and the limit of W$_{Gn}$.

4. GRAPH SAMPLING THROUGH KRONECKER DOUBLE DOVER

The difference between networked data and independent data samples is the topology types. Graph sampling needs to keep not only the distribution of vertices and edges but also the original topology of the data set, which can be verified through topology properties. For a data set with independent data samples, we only need to think about the distribution of the individual data points.

For different topology types, the majority of high degree vertices are in the core region and the majority of the low degree vertices are in the periphery region, except Erdos-Renyi random topology which is not real-world graph topology. Vertex hierarchy of these topology types is from core vertices which have more connections to periphery vertices which have fewer connections. The difference in the number of vertices in core and periphery categories is in an order of magnitude. Also, core vertices and periphery vertices have different meanings. Periphery vertices are less significant than core vertices.

Two factors affect the efficiency of the sampling process: one is the complexity of similarity comparison, which determines if it is possible to conduct the similarity comparison, and the other one is the impact of the merging methodology on the topology properties of the graph. Detailed proof and experiments will be shown in the following sections.

Graph Merging

For different topology types, we can divide vertices into two categories: core group and periphery group. When we do graph slicing and merging on different topology types, the sample set has to be meaningful and has to keep all of the graph properties. This can be illustrated in the following.

As shown in figure 5, we use Kronecker double cover to merge edges. In figure 5(a), a_1 and a_2 have two common neighbors which are b_1 and b_2 so a_1 and a_2 are similar. b_1 and b_2 are in two different graphs, $a_1b_1b_2$ and $a_2b_1b_2$. b_1b_2 are neighbors in graph $a_1b_1b_2$ and also in graph $a_2b_1b_2$, so that b_1 and b_2 are similar. W can give a_1, a_2, b_1, b_2 new names as (1,1), (1,2), (2,1), (2,2). We can merge (1,1) to (1,2) and merge (2,1) to (2,2) and the result is shown in figure 5(c).

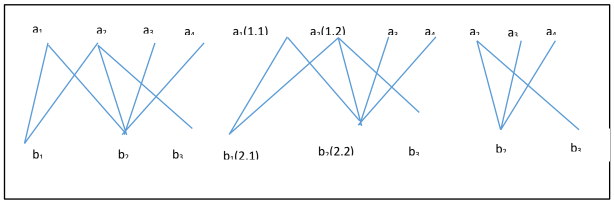


Fig. 5. Graph Merge through Kronecker Double Cover (a) a_1 and a_2 are similar because in graphs $_1b_1b_2$ and $a_2b_1b_2$, they have two vertices, b_1 and b_2, in common. Vertices b_1 and b_2 are similar because they appear together in more than one graph (graph $a_1b_1b_2$ and graph $a_2b_1b_2$) (b) a_1b_1 and a_2b_1 are merged. (c) a_2b_1 and a_2b_2 are merged.

```
Algorithm. Kronecker Double Cover

Input: G: original Graph, V(G): vertex set of graph G, E(G): edge set of graph G, T: Cutoff Threshold, sim: Similarity Measure

Output: G': sample graph

(1)Periphery_set
P=$\{(v,e)|\ v \in V(G), e \in E(G), degree(v) < T\}$
(2)Core_set C=G – P
(3)Kronecker set K={ $(v, w_1, w_2)|\ v \in V(P), w_1 \in V(P), w_2 \in V(P), (v, w_1) \in E(P), (v, w_2) \in E(P)\}$
(4)Kronecker similar groups S = $\{(count, w_1, w_2)|count = len([v_s])\ for\ ([v_s], w_1, w_2)\ in\ K\}$
(5)For sim = S.count.min( ) to S.count.max( )
    (5.1) $Frequent\ set\ F = \{(v, w_1, w_2)|(v, w_1, w_2) \in K, count(w_1, w_2) \geq sim\}$
    (5.2) Merged_set $M = \{(v, w)|if\ ([v_p], w_i, w_j) \in F, v\ replaces [v_p]$, w replaces $w_i\ and\ w_j\}$
    (5.3) V' ={$v|v \in V(M + C)\}$
    (5.4) E' ={$e|e \in E(M + C)\}$
    (5.5) V = V'
    (5.6) E = E'
```

On line 1, we generate Periphery set P by selecting all of the vertices with degrees less than T and also adding the edges between these vertices. In the Core set, C is the rest of the vertices and edges in the graph. On line 3, we generate Kronecker graphs K by choosing any two neighbors (w_1 and w_2) of a vertex v. On line 3, we count the graph frequency of the Kronecker graphs with two neighbors (w_1 and w_2) in common and save results to the frequent set F(v, w_1, w_2). On line 4, we create Kronecker similar groups based on the frequency of the isomorphic graphs. This is based on the assumption that the vertices and edges in periphery sets are end vertices and end edges. If they are by chance connected, those connections are random. We may not exactly know which vertices belong to the periphery set, but vertices with low degrees and less isomorphic graphs are highly likely in the periphery set. When we merge vertices and edges by choosing the frequency of the isomorphic graphs from low to higher up, we can check changes in the graph properties and topology properties in order to ensure the change is within a small range or can converge to a particular value. A merged set is a list of dictionaries to map vertices to new names. On line 5.3 through line 5.6, we update Kronecker graph set, vertex set, and edge set with new names in the merge set.

Based on Theorem 1, vertices need to have at least two neighbors to become constructible, otherwise, we call them **residuals**. Some vertices are constructible but the frequency of the isomorphic groups they can form is less than the similarity threshold. After merging, some vertices reduce the number of degrees and become residuals. These vertices are not removed from the graph, because we don't want to cut any connections in the graph and but, since these vertices have low degrees and less connections, they are not significant.

The Complexity of Kronecker Double Cover

Kronecker graph can also be presented in matrix format. Kronecker product and Kronecker double cover can be conducted through matrix operations, such as matrix multiplication and matrix factorization.

Lemma 1. The join of Kronecker graphs has a limited graphon.

Proof. Given a Kronecker graph, we can generate a sequence of graphs through Kronecker graph join. Based on the Existence of the Limit Theorem every convergent sequence of graphs or graphons has a limit graphon. Therefore, the sequence of graphs generated through Kronecker join has a limit graphon. ■

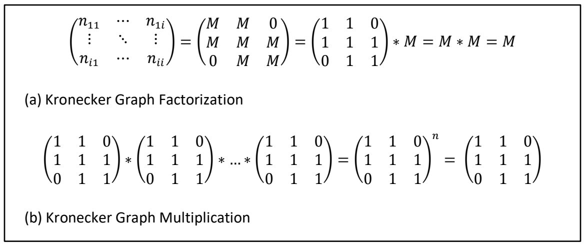
Lemma 2. Kronecker graph properties hold during projection.

Proof. Given a sequence of Kronecker graphs {G| G = G$_i$, i = 1, … , n}, the mathematical form of the sequence of graphs can be written as equation (9) below:

$$\prod_{i=1}^{n} G_i = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} * \ldots * \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix}$$

We illustrated this process in figure 6 (b). We can conclude that if any graph can be decomposed into a sequence of Kronecker graphs, the Kronecker properties hold during projection. ■

Lemma 3. Kronecker graph can be used to generate any binary graph with no self-loops and multiple edges.



$$\begin{pmatrix}n_{11}&\cdots&n_{1i}\\\vdots&\ddots&\vdots\\n_{i1}&\cdots&n_{ii}\end{pmatrix} = \begin{pmatrix}M&M&0\\M&M&M\\0&M&M\end{pmatrix} = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} * M = M * M = M$$

(a) Kronecker Graph Factorization

$$\begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} * \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} * \ldots * \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix}^{n} = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix}$$

(b) Kronecker Graph Multiplication

Fig. 6. Kronecker Graph Join and Factorization

Proof:

Given a Kronecker graph, we can construct any binary graphs without self-loops and multiple edges.

Kronecker graphs can be represented with matrix M

$$M = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} \qquad (10)$$

Now, if we multiply two Kronecker graphs, we can have

$$N = M * M = \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} * \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} =$$

$$\begin{pmatrix} \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} & \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} \\ \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} & \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} & \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} \\ \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} & \begin{pmatrix}1&1&0\\1&1&1\\0&1&1\end{pmatrix} \end{pmatrix}$$

In this way, as shown in figure 6 (a), we generate-graph N from graph M. ■

Homomorphism density can be used to estimate the space complexity of a sequence of graphs. When we decompose a graph into a set of Kronecker graphs, the space complexity can be estimated with lemma 4, as shown below. The complexity of $\binom{G}{k_{1,2}}$ are O(n^3) and the complexity is O(n^3) so that the homomorphism density of the Kronecker graphs is O(1).

Lemma 4. (Homomorphism Density of Kronecker graph)

Proof: Given a Kronecker graph G$_1$ = k$_{1,2}$, the number of copies of Kronecker graphs hom(k$_{1,2}$, G) is equation (11):

$$\hom(k_{1,2}, G) = \binom{2}{1} * \binom{1}{1} * \binom{G}{k_{1,2}} = 2 * \binom{G}{k_{1,2}}$$

The homomorphism density t(k$_{1,2}$, G) is equation (12):

$$t(k_{1,2}, G) = \frac{\hom\,(k_{1,2},G)}{|V(G)|^{|V(k_{1,2})|}} = \frac{2*\binom{G}{k_{1,2}}}{|V(G)|^3}$$

In which k$_{1,2}$ is a bipartite graph with one vertex on one side and two vertices on the other side. This lemma can also be applied to bipartite graphs K$_{1,n}$.

5. EXPERIMENTS

We used the Email-EU communication network data set to evaluate the performance of Kronecker double cover sampling methodology and compare it with EN-i [1] – an edge sampling methodology since Kronecker double cover is an edge sampling method as well. The measurements we choose to evaluate the quality of the sample sets are degree rank, degree frequency, and eigenvalue plot because Kronecker double cover operation merges edges so that the degree distribution can be potentially affected.

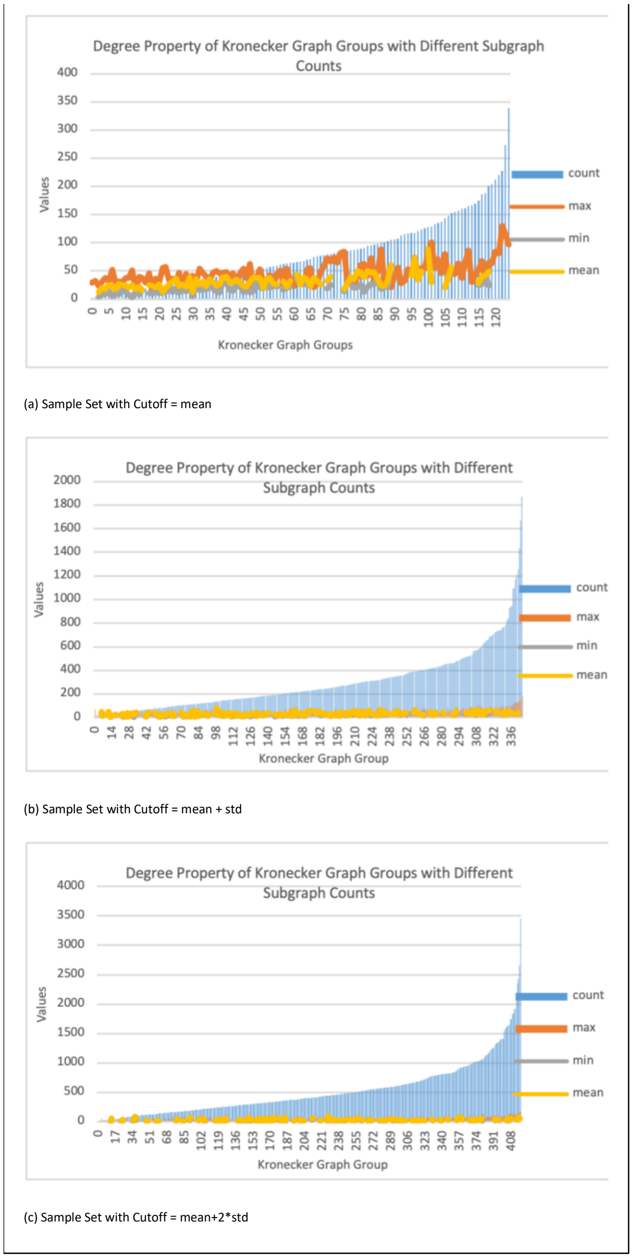
The first step is to split the original data set into two subgraphs – core and periphery, based on degree distribution. Since we only need to merge edges in periphery subgraphs. Vertex degrees belong to exponential distribution so that there is a clear gap between high degree vertices and low degree vertices. We generate sample sets with three different cutoffs which are *mean* - the average of the degrees, *mean+std* - the average of the degrees plus standard deviation, and *mean+2*std* - the average of the degrees plus 2 times the standard deviation. Since the number of degrees follows an exponential distribution, the standard deviation is normally greater than the mean of the number of degrees so we cannot use the average of degrees minutes standard deviation as the threshold for core and periphery separation.

The second step is to pick a Kronecker graph similarity threshold to find similar graphs. For experimental purposes, we choose three similarity thresholds which are 2, 3, and 4. When the similarity threshold increases, fewer edges can be merged but the graph size shrinks faster.

In figure 7, we group subgraphs based on similar Kronecker graph counts visualized the similar Kronecker graph count from low to higher up, and also visualized maximum degree, minimum degree, and an average degree in each group. Figure 7 (a)(b)(c) shows that similar Kronecker graph counts and vertex degrees are independent measurements. When similar Kronecker graph frequency increases in the reach group, the maximum, minimum, and average degrees in each group are random. During Kronecker graph double cover operation, the number of edges is reduced but it does not directly affect the degree distribution of the graph.

We merge Kronecker graphs from low-frequency groups to higher up, based on Kronecker graph similarity. This process converges within several iterations, as shown in figure 8. For the sample set with cutoff equal to *mean*, we have groups with similarity frequency from 2 to 339 and converged at 70. For the sample set with a cutoff equal to *mean+std*, we have groups with similarity frequency from 2 to 1874 and converge at 50. For the sample set with a cutoff equal to *mean+2*std*, we have groups with similarity frequency from 2 to 3461 and converge at 300. When cutoffs increase, similarity frequency converges faster. For cutoff equal to mean+2*std, some core vertices are included in periphery subgraphs which makes it converge slower.

During Kronecker similar graph merging, no connections are cut. When similar vertices and edges are merged, their neighbors are merged as well. After merging similar Kronecker graphs in periphery subgraphs, redundant connections between periphery vertices are merged. Based on topology types, as shown in figure 1, other than Erdos-random graphs, random connections between periphery vertices can be ignored. In order to maintain the connectedness of the graph, we merge them.

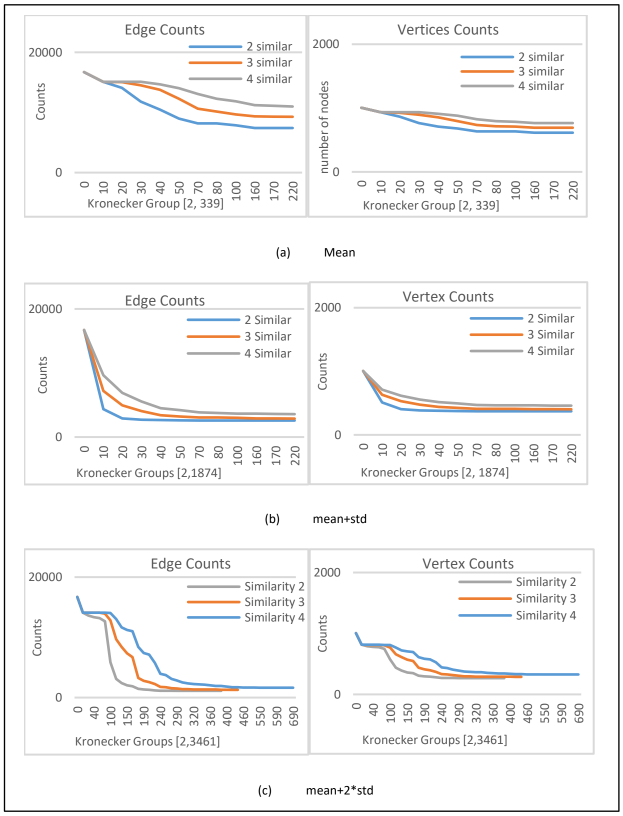


(a) Sample Set with Cutoff = mean

(b) Sample Set with Cutoff = mean + std

(c) Sample Set with Cutoff = mean+2*std

Fig. 7. Degree Property of Different Sample Sets



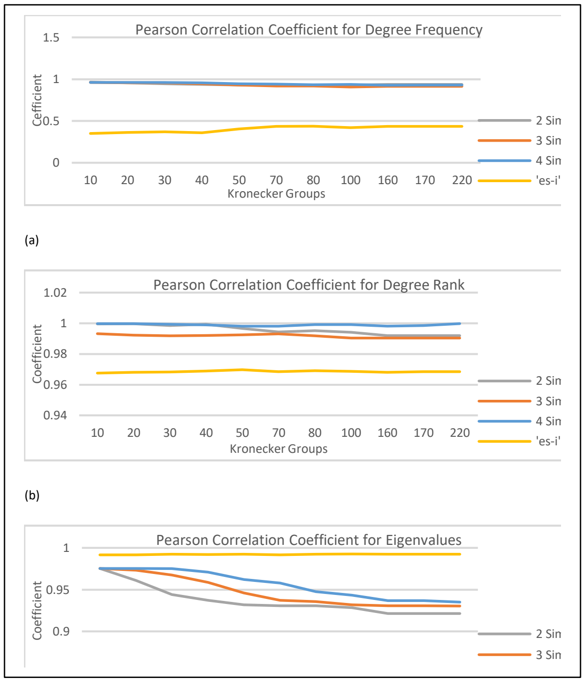
(a) Mean

(b) mean+std

(c) mean+2*std

Fig. 8. Edge and Vertex Sizes During Graph Merging

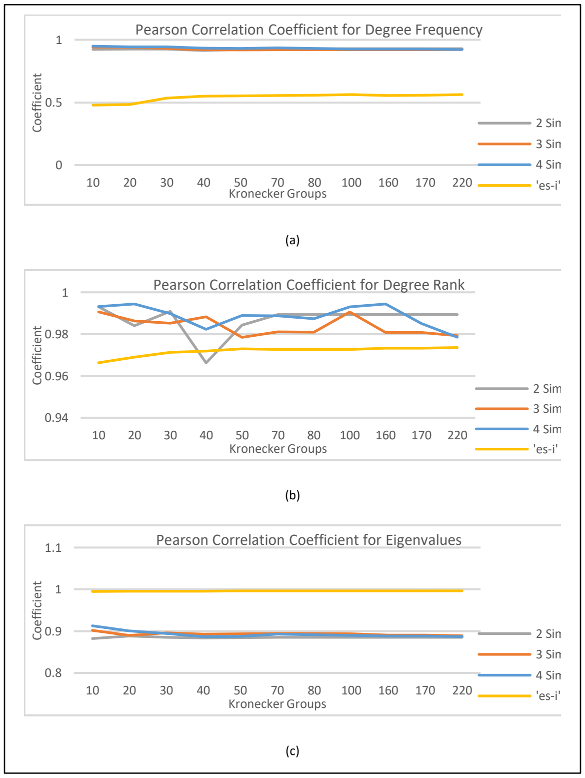
For degree rank and degree frequency, Kronecker double cover sample sets have higher Pearson correlation coefficients than ES-i sample sets with the same number of edges, as shown in figures 9(a)(b), 10(a)(b), 11(a)(b), which means that edge merging performs better than random edge sampling. Based on graph topology, degrees are not randomly distributed. When we equally select random edges from both core and periphery to form sample sets, degree property is changed. When cutoffs increase from *mean* to *mean+2*std*, the graph sizes shrink faster and there are less vertices to compare. For eigenvalue plot, as shown in figures 9(c),10(c), and 11(c), ES-i sample sets perform better, because, for the vertices in seed edge set, ES-i sampling method adds the rest of their connections to the sample set, that can better maintain the connectedness of the vertices in the sample set.



(a)

(b)

Fig. 9. Pearson Correlation Coefficients Between Original Set and Sample Sets Generated with Splitting Degree equal to *mean*



(a)

(b)

(c)

Fig. 10. Pearson Correlation Coefficients Between Original Set and Sample Sets Generated with Splitting Degree equal to *mean+std*

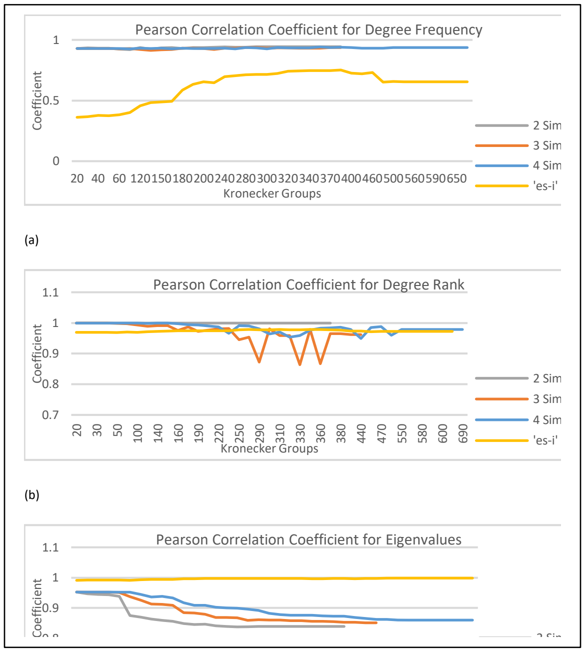


Fig. 11. Pearson Correlation Coefficients Between Original Set and Sample Sets Generated with Splitting Degree equal to *mean+2*std*

For Kronecker double cover sampling, Pearson correlation coefficients for degree ranks and degree frequencies are between 95% and 100%, as shown in figures 9(a)(b) and 10(a)(b), and Pearson correlation coefficients for eigenvalue plot are between 90% and 100% when cutoffs are set to mean and mean+std, as shown in figure 9(c) and 10(c). When the cutoff is set to mean+2*std, some core vertices are included in periphery subgraphs which causes merging between core and periphery vertices and make the connectedness of sample sets different from the original. The eigenvalue plot decreases from 95% to 85, as shown in figure 11(c). Degree ranks go up and down between 88% and 100%, as shown in figure 11(b), but degree frequencies are not affected.

The above experimental results also confirm the existence of the topology types and present the graph properties for different topology types.

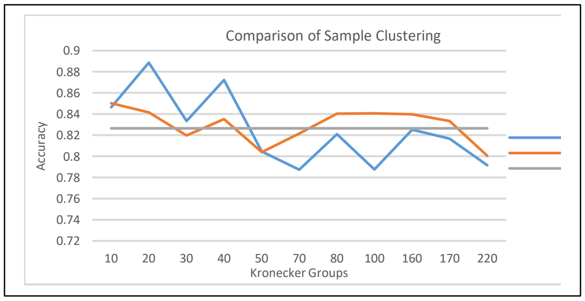


Fig. 12. Clustering Conductance on Different Email Eu Data Sets

We evaluate the sample quality on sample sets generated with a cutoff equal to the *mean*. As shown in figure 12, we visualized the clustering accuracy of Kronecker sample sets, ES-i sample sets, and the original data set. The number of edges started converging when Kronecker similarity frequency is around 50, as shown in figure 8(a). When the Kronecker similarity frequency is below the converging point, Kronecker sample sets have better accuracy than the original and ES-I sample sets with the same number of edges. When the Kronecker similarity frequency is beyond a converging point, ES-i sample sets have better accuracy than other sets. However, the degree ranks, degree frequencies, and eigenvalue plots decrease, when the Kronecker similarity frequency increases. In other words, when Kronecker similarity frequency is below the converging point, the sample quality increases when Kronecker groups are merged from low frequency to higher up. When Kronecker similarity frequency is beyond the converging point, the connectedness and graph properties are changed.

6. CONCLUSION

We present a new network sampling methodology based on Kronecker graph double cover. We theoretically proved that the network sampling through graph decomposition and reconstruction is feasible in terms of space complexity and time complexity, and can reserve graph properties and graph topology. To keep the connectedness of the network, graph sampling is through merging similar vertices and edges. Real-world graphs barely happen to be Erdos-random graphs. The topology types normally have a hierarchical structure. Kronecker double cover operation is different from degree reduction in that it can be used to merge vertices and edges but the selected vertices vary on the number of degrees.

In order to preserve the topology types of the graphs, we evaluate several different ways to separate core vertices and periphery vertices. We compare the degree rank and degree frequency between the original data set and the sample sets with the Pearson correlation coefficient. Sample sets and original data sets are similar, and the graph properties and graph topologies are preserved. We also compared the quality of the original data set and the sample set through graph clustering. After sample selection, we can efficiently improve the quality of the communities and shrink the size of the communities as well.

In the future, we are interested in better similarity measurements to catch the characteristics of the graph properties, especially to simulate the evolution of the graphs.

7. ACKNOWLEDGEMENT

Acknowledgement needs to be made to Venkata Jaipal Reddy Batthula, Ph.D. in Computer and Information Sciences of University of Arkansas at Little Rock (UALR) for performing peer-editing for final submission of this paper.

8. REFERENCES

[1] N.K. Ahmed., J. Neville, and R. Kompella, 2013. Network Sampling: from static to streaming graphs. ***ACM Trans. Knowl. Discov. from Data*.** Vol. 8, No. 2, Article 7 (June 2014),pp. 1-56. https://doi.org/10.1145/2601438

[2] R. Albert and A. Barabasi, Statistical Mechanics of Complex Networks. ***Reviews of Modern Physics***. Vol. 74. January 2002.

[3] V. Anand, P. Mehrotra, D. Margo, and M. Seltzer, Smooth Kronecker: Solving the Combing Problem in Kronecker Graphs. ***GRADES-NDA'20***, June 14, 2020. Portland, OR, USA.

[4] F. Belletti, J. Anderson, K. Lakshmanan, N. Mayoraz, P. Kanwar, T. Robie, T. Oguntebi., W. Krichene, and Y. Chen, Randomized Fractal Expansions for Production-Scale Public Collaborative-Filtering Data Sets. **arXiv e-prints 2019.**

[5] F. Belletti, K. Lakshmanan, W. Krichene, Y.-F. Chen, and J. Anderson, Scalable Realistic Recommendation Datasets through Fractal Expansions**, *arXiv* e-prints 2019**.

[6] D. Chakrabarti, Y. Zhan, and C. Faloutsos, R-MAT: **A** Recursive Model for Graph Mining. ***Proceedings of the 2004 SIAM International Conference on Data Mining*. 2004**, pp. 442-446.

[7] M. Faloutsos, P. Faloutsos, C. Faloutsos. On Power Law Relationships of the Internet Topology**. *SIGCOMM*** *1999.*

[8] M. Farzan, D. A. Waller, Kronecker Products and Local Joins of Graphs**. *Can. J. Math***., Vol. XXIX, No. 2. 1977, pp. 255-269.

[9] L.A. Goodman, Snowball Sampling. ***The Annals of Mathematical Statistics*,** pp. 148–170, 1961

[10] J. Gross**. *Topics in Graph Theory*.** http://www.cs.columbia.edu/~cs6204/files/Lec5-Automorphisms.pdf

[11] J. Lauri, and R. Scapellato. ***Topics in Graph Automorphisms and Reconstruction*.** Cambridge University Press. March 17, 2003. ISBN-9781316610442

[12] R. L. Hemminger, The group of an X-Join graphs. ***Journal of Combinatorial Theory 5***, 1968. pp. 408-418.

[13] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, Mathematically Tractable Graph Generation and Evolution, using Kronecker multiplication. ***Knowledge Discovery in Databases: PKDD 200*5,** pp 133-145.

[14] W.L. Hamilton. R. Ying, J. Leskovec, Inductive Representation Learning on Large Graphs. ***31st Conference on Neural Information Processing Systems (NIPS 2017)***, Long Beach, CA, USA.

[15] D. P. Sumner, Graphs Indecomposable with respect to the x-join. ***Discrete Mathematics*** Vol. 6(1973) pp. 281-298.

[16] D. A. Waller, Double Covers of Graphs**. *Bull. Austral. Math. Soc*.** Vol. 14 (1976), pp. 233-248.

[17] Y. Zhao. Graph Theory and Additive Combinatorics**. *Notes for MIT 18.217*** (Fall 2019). pp. 49-58. http://yufeizhao.com/gtac/