Temporal Graph Generation Based on a Distribution of Temporal Motifs

Sumit Purohit
Pacific Northwest National Laboratory
Richland, WA
Sumit.Purohit@pnnl.gov

Lawrence B. Holder
Washington State University
Pulllman, WA
holder@wsu.edu

George Chin
Pacific Northwest National Laboratory
Richland, WA
George.Chin@pnnl.gov

ABSTRACT

Generating a synthetic graph that is similar to a given real-world graph is a critical requirement for privacy preservation and benchmarking purposes. Various generative models attempt to generate static graphs similar to real-world graphs. However, generation of temporal graphs is still an open research area. We present a temporal-motif based approach to generate synthetic temporal graph datasets and show results from three real-world use cases. We show that our approach can generate high fidelity synthetic graph. We also show that this approach can also generate multi-type heterogeneous graph. We also present a parameterized version of our approach which can generate linear, sub-linear, and super-linear preferential attachment graph.

KEYWORDS

Temporal Graph, Graph Generative Model,Motifs Distribution

1 INTRODUCTION

Graphs are a natural and flexible representation of a set of entities and the relationships among them. Static graph represents a set of objects and a set of pairwise relations between them. A temporal graph is a generalization of a static graph which changes with time. Time can also be modeled as a vertex or edge label, which makes temporal graphs a special case of attributed graphs. Incorporating time into the static graphs has given rise to a new set of challenging and important problems that cannot be modeled as a static-graph problem [15]. Many domains such as social networks, communication, transportation, sensor networks, biological networks, co-authorship networks, and procurements can be naturally modeled as temporal graphs.

Many graph generative models are studied and developed to generate synthetic graphs. The Random Model [7] and the Preferential Attachment Model [2] are classic graph generative models. The Chung-Lu model provides a random model to generate power law graphs [1] using an input degree distribution. Recently Leskovec and Foloutsos [13, 14] presented the Kronecker model based on Kronecker matrix multiplication to generate syntactic graphs that replicate multiple graph properties. It uses maximum-likelihood approach to estimate up to four parameters to model the graph. For a graph with N nodes, the likelihood has contributions from N! permutations of the nodes[14]. Gleich et al. [9] present method-of-moments estimators that are computationally much simpler than maximum likelihood. The Block Two-Level Erdős-Rényi (BTER) model [20] assumes that any graph with a heavy-tailed degree distribution and community structure must contain a scale-free collection of dense ER subgraphs. The BTER model *implants* a fixed amount of communities in the network and then generates dense ER subgraphs within the community. All such models attempt to satisfy some global graph properties and replicate a given degree distribution, but do not guarantee the preservation of localized structural properties. Many of these models also do not generate temporal graphs and can only generate homogeneous static graphs.

Small subgraph patterns in networks, called network motifs or graphlets are crucial indicators of the structure and the evolution of the graphs [17]. Paranjape et al. [17] develop a notion of a temporal network motif as an elementary unit of temporal networks and provide a general methodology for counting such motifs. Pržulj, Nataša [18] postulates that global statistics on some data may be substantially biased, or even misleading with respect to the (currently unknown) full network at certain point of time. Conversely, certain neighborhoods of these networks are well studied, and so local statistics applied to the well studied areas are more appropriate. [18] presents new systematic measures of a network's local structure that imposes a large number of similarity constraints on networks and use it to compare two cellular networks. It presents 73 graphlet degree distributions of graphlets with two to five nodes each. Sarajlic et al. uses directed graphlets to generalize network distance measures to compare directed networks [19].

This research presents a graph generative model that uses non-overlapping directed temporal motifs. Theses motifs are easy to compute and can model wide range of temporal networks. This approach strives to preserve local temporal

structures while generating synthetic graphs. These easy to compute temporal atomic motifs are used to define any real-world graph. The core hypothesis of this research is that preserving local temporal-motifs is sufficient to generate synthetic graphs that also exhibit similar global graph properties of the corresponding real-world graph.

Section 2 presents all the temporal definitions used in the research. Section 3 describes Structural Temporal Modeling (STM) approach to characterize a real-world graph using set of algorithms. Graph (non) linearity is discussed in Section 4. Section 5 presents empirical results using real-world graphs to show the performance of STM to generate synthetic graphs. Multi-type graphlet distribution is discussed in section 6. Conclusion, Future work, and Acknowledgment are described in sections 7,8,and 9 respectively.

2 PRELIMINARIES

This section provides formal definitions of temporal graph, temporal edge, and temporal motifs to describe research methodology.

Definition 1: Temporal Graph : A temporal graph is a directed graph that can be represented as 5-tuple. G = (V, E, L, l,T), where V is set of vertices, E is set of edges such that E $\subseteq$ V x V, L is set of labels on vertices and edges, and l is a function V $\cup$ E $\rightarrow$ L that assigns labels to vertices and edges, and T is function that assign time-stamp *t* to each edge e $\subseteq$ E.

Definition 2: Temporal Edge: A 3-tuple (v_s, v_d, t_k) defines a temporal edge e_k between vertex v_i and v_j at time t_k.

Definition 3:: Temporal Ordering: A temporal ordering of temporal edges *e_1 , and e_2* where *e_1 is defined as (v_{s1}, v_{d1}, t_1) and e_2* as (v_{s2}, v_{d2}, t_2) is defined that *$e_1 > e_2$* iff:

- $t_1 < t_2$
- $t_1 = t_2$ and $v_{s1} < v_{s2}$
- $t_1 = t_2$, $v_{s1} = v_{s2}$ and $v_{d1} < v_{d2}$

Temporal ordering of vertices is defined based on *vertex-birth-time* defined below. A vertex v_{s1} is considered smaller than v_{s2} iff *vertex-birth-time*(v_{s2}) > *vertex-birth-time*(v_{s1}). Vertex Id is also an easy to preserve alternative to the *vertex-birth-time* and can be used instead of it for the vertex ordering.

Definition 4: Atomic Motif: For this research an atomic motif *m* is defined as a subgraph with up to three edges and three vertices without self-loop.

Figure 1 shows structural atomic motifs used in this research. These motifs are selected because they are the smallest possible motifs that can represent different varieties of real-world graphs. These motifs are also simple enough to be computed for large graphs of billion edge scale.

Definition 5: Temporal Atomic Motif: A temporal atomic motif m_{sk} is a subgraph with temporal ordering of edges and vertices.

Figure 2 represents all 22 temporal atomic motifs generated from six atomic motifs described above.

<m_{10}>, <m_{11}>, <m_{12}>, <m_{13}>
.....
.....
<m_{40}>, <m_{41}>, <m_{42}>, <m_{43}>
<m_{50}>, <m_{51}>, <m_{52}>
<m_{60}>, <m_{61}>, <m_{62}>

It shows that a three-vertex motif m_1, m_2, m_3, and m_4 can have four variations each based on how many vertices are already generated in the temporal graph at a given time t, and how many of them are new vertices going to be added to the temporal graph because of the motif. These four variations are when zero,one,two,or three vertices are re-used (or newly created).

Definition 5: Temporal Edge Arrival Probability: Temporal edge arrival probability is defined as the sum of edge arrival likelihood from all the temporal atomic motifs.

The first two atomic structural motifs have three edges each and every addition of that motif will increase the graph size by 3. Structural motifs m_3, m_4, and m_5 add two edges to the graph, and structural motif m_6 which represents a single directed edge adds one temporal edge to the graph.

$$p(e) = \sum_{m=1}^{2}\sum_{k=0}^{3} 3 * p(m_{mk}) + \sum_{m=3}^{5}\sum_{k=0}^{2} 2 * p(m_{mk}) + \sum_{k=0}^{1} p(m_{6k})$$

Definition 6: Temporal Vertex Arrival Probability: Temporal vertex arrival probability is defined as the sum of vertex arrival likelihood from all the temporal atomic motifs. Every atomic motif is expanded to (n+1) temporal motifs where n is the number of vertices in the atomic motif.

$$p(v) = \sum_{m=1}^{4}(3 * p(m_{m0}) + 2 * p(m_{m1}) + p(m_{m2})) + \sum_{m=5}^{6}(2 * p(m_{m0}) + p(m_{m1}))$$

3 STRUCTURAL TEMPORAL MODELING

We define Structure Temporal Modeling (STM) as a process of identifying temporal motifs in a real-world graph. We

define some easy to compute atomic motifs such as those shown in Figure 1 which can characterize any given real-world graph. We search for these motifs in the graph starting from larger motifs to smaller ones. Once larger motifs are found we update the graph by removing corresponding edges from the graph. This strategy guarantees that the motifs are found in a mutually exclusive fashion and we do not include overlapping motifs. This approach is similar to the Minimum Image Support (MIS) approach [3, 4, 22] used in graph mining research. MIS defines the support of a pattern as the minimum number of distinct mappings for any vertex over all the pattern instances in the graph. As pattern size grows, the MIS values of larger patterns are never greater than the MIS of smaller sub-patterns. Similarly, any edge can only participate in one non-overlapping motif, and the number of such motifs is always bounded by the distinct edge set in a given temporal motif.

We also compute the information content of every atomic motif by identifying the number of new vertices used in the atomic motif formation. This leads to the generation of *temporal atomic motifs* as defined above. The number of temporal motifs is always (n+1) where n is the number of vertices in the atomic motif. We define *vertex-birth-time* of a vertex as the earliest arrival time of temporal edges associated with this vertex. We also define *motif-birth-time* as the earliest time at which any edge of that motif arrives. Using these two definitions we compute the information content of a motif as the number of new and old vertices associated with the motif. This leads to multiple *temporal atomic motifs* for a given *atomic motif*. For example, in Figure 1 a triangle atomic motif is expanded to four temporal atomic motifs where 0,1,2, or 3 vertices are new (or re-used). The six atomic motifs in Figure 1 can generate up to 22 temporal atomic motifs as shown in Figure 2.

For each temporal atomic motif we also compute its *formation time* which is the total time taken by the motif to fully form. It is computed as the difference in the earliest and latest edge associated with the motif. At the same time, we also compute the *average arrival delay* in generating each edge of the motif. *average arrival delay* allows us to distinguish between rapidly formed motifs and the ones that form over a longer period of time.

Distribution of such temporal atomic motifs is computed for a given real-world graph. Motif *arrival rates* are computed by normalizing the distribution over the entire duration of the input graph. This normalized distribution is used to generate its synthetic version and the same distribution is also computed for the synthetic graph. Variation in these two distributions is used as a metric to assess the quality of the synthetic graph.

The generator component of the STM uses the distribution to iteratively generate all the temporal motifs using arrival

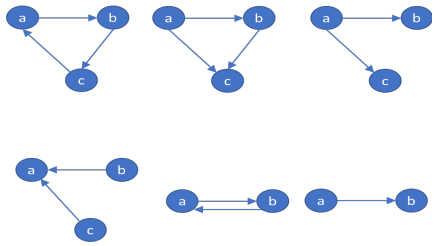

Figure 1: Atomic Motifs

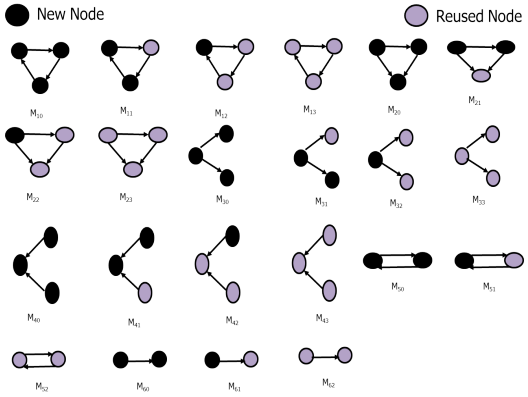

Figure 2: Atomic Temporal Motifs

rates as *generation probabilities*. STM uses the information content of the motifs to decide whether to create new nodes or reuse existing nodes in the graph at a given point of time. STM also uses *formation time* and *average arrival rate* to delay the formation of the temporal-motif. This is critical to preserve temporal evolution of the local structures. We have developed a queue based scheduler to delay the generation of a specific edge in the motif.

STM Algorithm

In this section we illustrate the basic algorithm used to compute temporal motifs distribution from a real-world graph. Algorithm 1 is the entry point of STM and it takes input graph G and a predefined ordered list of structural atomic motifs M as an input. It iterates through all the structural motifs and computes arrival rates of each temporal motif in the input graph. Arrival rate of a temporal motif is used as a generative probability in the generation step. Arrival rate is normalized using temporal span of the input graph.

Algorithm 2 takes current graph, a structural motif, and temporal span of the real-world graph as an input and computes arrival rates of all corresponding temporal motifs. Current implementation uses graphframe based Domain-Specific Language (DSL) for expressing structural motifs. For example $(a) - [e1] - > (b); (b) - [e2] - > (c); (c) - [e3] - > (a)$

Algorithm 1: STM(G,M)

Data: G,M
Result: P_m
1 initialize P_m with default value zero
2 d=getTemporalDuration(G)
3 **for** *each* $m \in M$ **do**
4 | <G,P_m> = getTemporalMotifs(G,m,d)
5 **end**
6 **return** P_m

Algorithm 2: GetTemporalMotifs(G,m,d)

Data: G,m,d
Result: G_{new}, P_m
1 $I_a \leftarrow$ find(G,m)
2 $I_n \leftarrow$ getNonOverlapping(I_a)
3 $P_m(m) \leftarrow |I_a|$/d
4 $P_m \leftarrow P_m \cup P_m(m)$
5 G_{new} = updateGraph(G,I_n)
6 **return** <G_{new}, P_m>

describes a triangle with three edges *(a,b), (b,c), and (c,a)*. We add additional attribute *time* to represent temporal edges in the input graph. Line 1 identifies all the structure motifs in the graph. Line 2 filters them based on the non-overlapping temporal constraints. I_n represents a set of temporal motifs of a given atomic motif. Line 3 computes arrival generative probability of all such motifs in the set I_n. Line 4 and 5 update the global motif distribution and the input graph by removing edges used in forming I_n.

Algorithm 3: getNonOverlapping(I_a)

Data: I_a
Result: I_n
1 $E_{es} \leftarrow$ getDisticntEdgeSet(I_a)
2 $e_{min} \leftarrow$ min(E_{es})
3 $I_c \leftarrow$ getCandidateMotifs(I_a,e_{min})
4 $e_v \leftarrow$ empty
5 $I_n \leftarrow$ empty
6 **for** *each* $i \in I_c$ **do**
7 | **if** $E(i) \notin e_v$ **then**
8 | | $I_n \leftarrow I_n \cup i$
9 | | $e_v \leftarrow e_v \cup E(i)$
10 | **else**
11 | | **continue**
12 | **end**
13 **end**
14 **return** I_n

Algorithm 3 takes all instance of overlapping motifs found in the graph and returns set of non-overlapping temporal instances. Line 1 computes E_{es} which is a set of *edge sets* where each *edge set* is a collection of distinct edges across E_{es}. This guarantees that no edge is used more than ones in the motif formation. Line 2 finds number of all the *edge set* and Line 3 finds all the candidate temporal motifs using the minimum edge set. Line 6 to 12 iterates through candidate motifs and identifies all the non-overlapping temporal motifs.

4 (NON)LINEARITY IN TEMPORAL GRAPHS

Preferential attachment and Power law are highly observed phenomena in real-world graphs of many domains. The classical preferential attachment model for networks by Barábasi and Albert [2] assumes a linear relationship between the number of neighbors of a node in a network and the probability of attachment. Kunegis et al. [12] perform an extensive study of forty-seven diverse Web network datasets from seven network categories and show that contrary to the usual assumption, preferential attachment is nonlinear in the networks under consideration. They also observe that the deviation from linearity is dependent on the type of network, giving sub-linear attachment in certain types of networks, and super-linear attachment in others.

Classical work on linearity of graphs does not consider temporal evolution of the graph. Many *Network Growth Models* are suggested [2, 6, 8, 10] to define a preferential attachment function to model growth of the graph. We propose to use an easy to compute power-law variant as a temporal graph growth model. We use the following probability density function to select a vertex in the existing temporal graph at a give point of time t=t_i.

$$f(x, \alpha) = \alpha * x^{\alpha-1}$$

When the new edges are added to the temporal graph, we also update the *scale* of the probability function simultaneously, if one or more new vertices are added to the graph. Figure 3 shows different α values and their impact on vertex degree distribution. We observe the following phenomena using the α parameter in STM:

- α = 1 generates a linear preferential attachment graph which corresponds to the Barábasi and Albert model of scale free networks. This is the default value used in STM without domain knowledge of the input graph.
- α > 1 generates a preferential attachment network with sub-linear function. We can generate an extremely sparse graph with very few hubs of small size. We also observe that the rate of *sparsification* plateaus for α > 1, depending on the graph domain.
- α < 1 varies between linear and super-linear graph. 0.5 < α < 1 successfully generates many real-world graph which follow the power law distribution such as

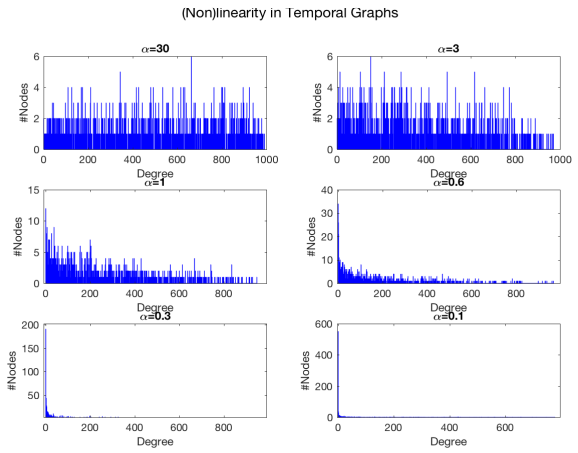

Figure 3: Alpha Variability

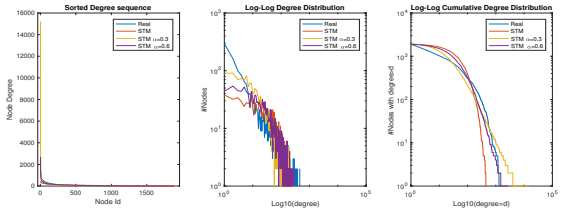

Figure 4: Synthetic CollegeMsg network generation

communication and social networks. $\alpha < 0.5$ quickly leads to a super-linear graph where a single vertex may acquire almost 100% of the edges asymptotically.

5 EXPERIMENTS

We have developed a scalable framework using Apache Spark [21] and GraphFrames [5] to compute the distribution of temporal atomic motifs. We have also developed a graph generator using Python (https://github.com/lbholder/graph-stream-generator) that takes the distribution as an input and generates a synthetic graph. We present results from two domains: social networks and financial network. We were able to model two million edge graphs successfully. We also used parameterized version of STM using different α values and as discussed in the section above $\alpha = 0.6$ generates synthetic graph closest to real-world graph.

Figure 4 shows the synthetic graph generation of a college messaging temporal network [16]. This dataset is comprised of private messages sent on an online social network at the University of California, Irvine. Users could search the network for others and then initiate a conversation based on profile information. An edge (u, v, t) means that user u sent a private message to user v at time t.

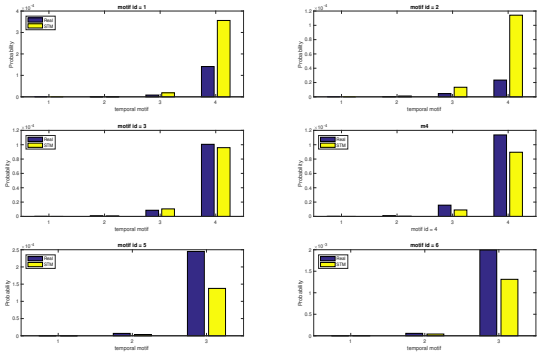

Figure 5: CollegeMsg Temporal Motif Distribution

	|V|	$\|E_{temporal}\|$	$\|E_{static}\|$	Time
CollegeMsg	1899	59,835	20,296	193 days
Bitcoin Alpha	3,783	24,186	24,186	1901 days
PhoneEmail	986	20,001	14,613	365 days

Table 1: Temporal Graphs Properties

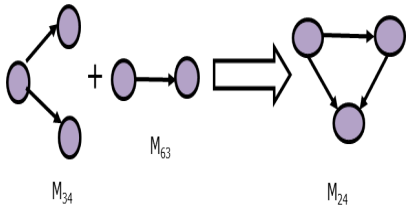

Figure 6: Motif formation using existing vertices

Figure 4 shows that STM closely follows the degree distribution curve of the real graph. It also shows that STM does not create few very high degree nodes but rather creates smaller hubs. Figure 5 shows structural temporal motif distribution of the CollegeMsg network and its synthetic counterpart. Figure 5 combines each temporal motif by its structural parent and plots them separately to accommodate the probability range for this category. It shows similar motif distribution except for a few temporal motifs. Close analysis of the graph formation revealed that many of the single edge motifs m6 are subsumed by m3 and m4 which leads to formation of m1 and m2. This is only applicable to motifs which reuse existing vertices to form the motif. For example formation of m_{63} selects two existing nodes n1 and n2 from the graph and if n1 and n2 are part of existing $m3$ or $m4$ then the addition can results in the formation of $m1$ and $m2$. This explains higher probabilities of m_{14} or m_{24}. This phenomena is depicted in Figure 6.

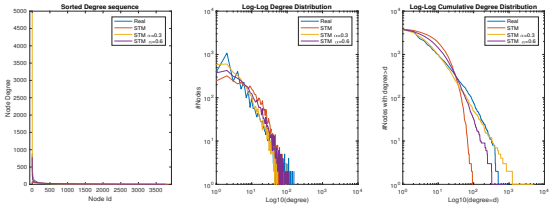

Figure 7: Synthetic Bitcoin network generation

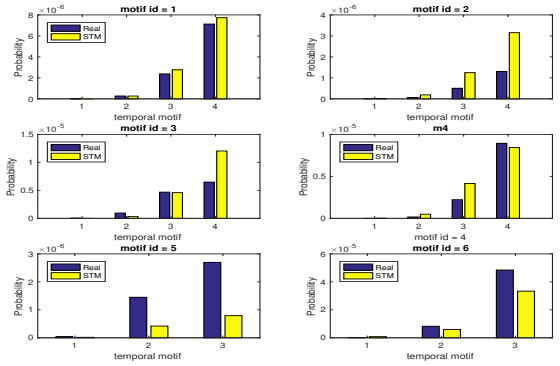

Figure 8: Bitcoin Temporal Motif Distribution

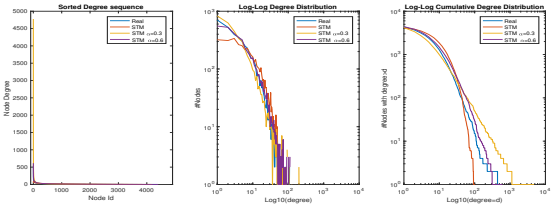

Figure 9: Multi-type Synthetic Phone-Email network generation

The second dataset is *who-trusts-whom* network of people who trade using Bitcoin on a platform called Bitcoin Alpha [11, 17]. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Members of Bitcoin Alpha rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. For this research work we do not model the rating attributes but only model the temporal evolution of the user transactions. Figures 7 and 8 shows the results.

6 MULTI-TYPE MOTIF DISTRIBUTION

Many existing synthetic graph generators can only generate homogeneous graph with single edge type. There is an opportunity to develop multi-type, multi-channel graph generator and we apply STM using multi edge-type motifs. We can generate two channels simultaneously using 114 different temporal motifs. We used PNNL's anonymized internal communication dataset snapshop that contains phone and email communication network within organization. Figure 9 shows a multi-type synthetic graph generation result using same set of α values.

7 CONCLUSION

Modeling temporal graphs and developing synthetic temporal graph generator is an active research area. Various graph generative models have been proposed that attempt to replicate a given degree distribution or preserve some global properties in a homogeneous static graph. We present a novel approach of using temporal motif distribution to generate synthetic temporal graph. This approach closely follows temporal evolution of the graph and preserve local structures within the graph. We present our results using two real-world homogeneous graphs. We also present that our approach can generate high fidelity multi-type, multi-stream temporal graph and we present our results using a real-world connected graph of two edge types. We also propose a simple power law variant to be used in temporal graph generation with α parameter. We also show that we can generate linear, sub-linear, and super-linear graph using different α values.

8 FUTURE WORK

Future work will compute local vertex attributes to use a machine learning model to select appropriate vertex in the temporal graph. We will address explosion of number of candidate temporal motifs in a multi-type graph by only modeling a subset of most informative temporal motifs based on data-driven domain knowledge. We will also develop new temporal metrics to measure the fidelity of synthetic graphs.

9 ACKNOWLEDGMENT

We thank the DARPA Modeling Adversarial Activity (MAA) program for funding this project under contracts HR0011728117, HR001178235, and HR0011729374. The associated PNNL project number is 69986. This work is also supported by the National Science Foundation under Grant No. 1646640. A portion of the research was performed using PNNL Institutional Computing (PIC) at Pacific Northwest National Laboratory.

REFERENCES

[1] William Aiello, Fan Chung, and Linyuan Lu. 2000. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. Acm, 171–180.

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[3] Björn Bringmann and Siegfried Nijssen. 2008. What is frequent in a single graph?. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 858–863.

[4] Sutanay Choudhury, Khushbu Agarwal, Sumit Purohit, Baichuan Zhang, Meg Pirrung, Will Smith, and Mathew Thomas. 2017. Nous: Construction and querying of dynamic knowledge graphs. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE.

[5] Ankur Dave, Alekh Jindal, Li Erran Li, Reynold Xin, Joseph Gonzalez, and Matei Zaharia. 2016. Graphframes: an integrated api for mixing graph and relational queries. In *Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems*. ACM, 2.

[6] Steffen Dereich, Peter Mörters, et al. 2009. Random networks with sublinear preferential attachment: degree evolutions. *Electronic Journal of Probability* 14 (2009), 1222–1267.

[7] Paul Erdos. 1959. On random graphs. *Publicationes mathematicae* 6 (1959), 290–297.

[8] Alan Gabel and Sidney Redner. 2013. Sublinear but never superlinear preferential attachment by local network growth. *Journal of Statistical Mechanics: Theory and Experiment* 2013, 02 (2013), P02043.

[9] David F Gleich and Art B Owen. 2012. Moment-based estimation of stochastic Kronecker graph parameters. *Internet Mathematics* 8, 3 (2012), 232–256.

[10] Hawoong Jeong, Zoltan Néda, and Albert-László Barabási. 2003. Measuring preferential attachment in evolving networks. *EPL (Europhysics Letters)* 61, 4 (2003), 567.

[11] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 221–230.

[12] Jérôme Kunegis, Marcel Blattner, and Christine Moser. 2013. Preferential attachment in online networks: Measurement and explanations. In *Proceedings of the 5th Annual ACM Web Science Conference*. ACM, 205–214.

[13] Jurij Leskovec, Deepayan Chakrabarti, Jon Kleinberg, and Christos Faloutsos. 2005. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 133–145.

[14] Jure Leskovec and Christos Faloutsos. 2007. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th international conference on Machine learning*. ACM, 497–504.

[15] Othon Michail. 2016. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics* 12, 4 (2016), 239–280.

[16] Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley. 2009. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the Association for Information Science and Technology* 60, 5 (2009), 911–932.

[17] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 601–610.

[18] Nataša Pržulj. 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23, 2 (2007), e177–e183.

[19] Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaveroğlu, and Nataša Pržulj. 2016. Graphlet-based characterization of directed networks. *Scientific reports* 6 (2016), 35098.

[20] Comandur Seshadhri, Tamara G Kolda, and Ali Pinar. 2012. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E* 85, 5 (2012), 056109.

[21] Apache Spark. 2016. Apache Spark: Lightning-fast cluster computing. *URL http://spark. apache. org* (2016).

[22] Carlos HC Teixeira, Alexandre J Fonseca, Marco Serafini, Georgos Siganos, Mohammed J Zaki, and Ashraf Aboulnaga. 2015. Arabesque: a system for distributed graph mining. In *Proceedings of the 25th Symposium on Operating Systems Principles*. ACM, 425–440.