Building Graphs at Scale via Sequence of Edges: Model and Generation Algorithms (Extended Abstract)

Yu Liu*
Beijing Jiaotong University
yul@bjtu.edu.cn

Lei Zou
Peking University
zoulei@pku.edu.cn

Zhewei Wei
Renmin University of China
zhewei@ruc.edu.cn

2022 IEEE 38th International Conference on Data Engineering (ICDE) | 978-1-6654-0883-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDE53745.2022.00145

***Abstract*—Real-world graphs exhibit many interesting properties that differentiate them from random graphs, which have been extensively studied for the past decades. For various proposed generative models, a majority of them build the graph by sequentially adding each node and the attached edges. However, the growth of many real-world graphs, such as social networks, is naturally modeled by the sequential insertion of edges. Unfortunately, to the best of our knowledge, no generative model has been proposed to reveal this process.**

We propose the first sequence-of-edges model, denoted as *temporal preferential attachment (TPA)*. It relies on *preferential attachment (PA)*, one of the most influential mechanisms to generate scale-free graphs, and takes time-decay effect and node fitness into consideration. Empirical analysis demonstrates that our model preserves several key properties of the real-world graphs, including both the properties observed from the snapshot graphs (e.g., power-law distribution) and temporal properties observed from the graph generation process (e.g., shrinking diameter). Meanwhile, our model is sufficiently general to accommodate several forms of time decay and fitness distributions. Then, we design two efficient algorithms that generate TPA graphs with billions of edges in several minutes.

I. Introduction

Graphs are widely used to model the relationships between objects in various applications, such as websites, social networks and knowledge graphs. Some of the real-world graphs, such as social networks and graph streams, exhibit structural properties that are fundamentally different from those of random graphs, e.g., the Erdős and Rényi's graphs. Tracing back to the pioneering work in early 20th century, considerable research has been devoted to the study of the properties of real-world graphs, with new observations and understandings continuously arising in the past ten years. These findings not only further our understanding of graph theory, but also change the way we design graph algorithms and systems. For example, the fact that the degrees of real-world graphs follow heavy-tailed distribution facilitates a number of efficient graph algorithms. On the other hand, the same property poses new challenges for graph-parallel systems.

The study of structural properties on real-world graphs can be broadly divided into three categories. First, many early works focus on making observations for various structural properties on real-world graphs. Second, based on these observations, a significant amount of research tries to propose complex graph models that explain the observed properties. Finally, to incorporate the growing need of large synthetic graphs, a few recent work focuses on designing efficient algorithms to generate synthetic graphs that look like real-world graphs. State-of-the-art algorithms can generate a billion-edge graph on a commodity machine [1], or a trillion-sized graph in the distributed environment. This not only avoids the privacy concerns of real-world data, but also facilitates the evaluation of algorithms and systems for large graphs.

Sequence-of-edges graph. As pointed by [2], growth is the very important property of real-world graphs. Nonetheless, graphs generated by various applications exhibit fundamentally different growth patterns. For citation networks and Wikipedia graphs, they evolve in a *node-centric* way, i.e., the graph expands by additions of nodes (e.g., papers, entities) and the attached edges. On the other hand, graphs such as social networks and communication networks grow in an *edge-centric* way. For example, social network grows by building new edges (e.g., friendship, the follow relationship) between nodes (i.e., users). However, each newly established interaction does not necessarily involve the addition of new users. In fact, interaction among existing users evolves over time and comprises a large fraction of edges in the network. Another example is the graphs that can be represented by a sequence of time-stamped edges, such as graph streams. It is natural to model these graphs by a *sequence-of-edges* manner.

Definition 1 (*Sequence-of-edges* graph)**.** *The generation process of a sequence-of-edges graph is defined as* $G = (G_1, \ldots, G_k, G_{k+1}, \ldots)$*, where*

$$G_{k+1}.V = G_k.V \cup \{u_{k+1}\} \cup \{v_{k+1}\},$$
$$G_{k+1}.E = G_k.E \cup \{(u_{k+1}, v_{k+1})\}.$$

For $k \in [1, \infty)$*, the graph grows by adding an edge* (u_{k+1}, v_{k+1}) *to the current graph* G_k*. Note that the two endpoints* u_{k+1} *and* v_{k+1} *do not have to be nodes in* $G_k.V$*, and can be newly added ones.*

In this paper, we aim to propose a sequence-of-edges model by non-trivially integrating several key ingredients of

*Work partly done at Wangxuan Institute of Computer Technology, Peking University.

2375-026X/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDE53745.2022.00145

the preferential attachment-based model, which is powerful enough to explain the well-recognized properties of real-world graphs, and yet simple enough so we can design highly scalable algorithms that generate large synthetic datasets.

II. TPA: Our Sequence-of-edges Model

A. Model Specification

We formally describe our temporal preferential attachment (TPA) model as follows. For simplicity, we focus on undirected graphs.

Step 1. Start with a small random graph (e.g., ER graph) G_k, which consists k vertices $\{v_1, \ldots, v_k\}$. Note that the subscript k represents the number of nodes in the current graph. Place one virtual node v_{k+1} *outside of G_k. For simplicity, we can set $k = 1$.*

Step 2. At each time, add one edge $e = (u_1, u_2)$ between nodes $\{v_1, \ldots, v_k, v_{k+1}\}$.

- *For each $v \in \{v_1, \ldots, v_k\}$, the node preference, denoted by $tpa(v)$, is computed by the attachment function. The preference of v_{k+1} is a constant $\alpha \in [1, \infty)$ given as model parameter.*
- *Both endpoints u_1 and u_2 of e is chosen in proportional to the node preference. As long as a self-loop is formed ($u_1 = u_2$), we re-sample u_2.*
- *If v_{k+1} is chosen as one endpoint of e, add it (and e) to the current graph, and place virtual node v_{k+2}. Otherwise, we only add the edge to the graph.*

The attachment function. We adopt a general function which consists three independent parts:

$$tpa(v) = f(d(v)) \cdot g(\Delta t(v)) \cdot h(v). \tag{1}$$

The degree-based PA function $f(\cdot)$ is a monotonic increasing function of node degree. By default, we set it as the power-law (i.e., polynomial) function: $f(d(v)) = d(v)^{\beta_d}$, where $\beta_d \in [0, \infty)$ is the preferential attachment exponent. Similarly, the temporal PA function $g(\cdot)$ parameterized by $\beta_t \in [0, \infty)$ is a monotonic decreasing function of node age. The function has a general form and can be power law, exponential ($g(\Delta t(v)) = e^{-\beta_t \Delta t(v)}$) or log-normal ($g(\Delta t(v)) = e^{-\beta_t \log^2(\Delta t(v)+1)}$). We assume nodes are numbered $v_1, \ldots, v_k, \ldots$ by their insertion order. By supposing *nodes are inserted sequentially and at a steady rate*, the time decay of node v_k at time t is $t-k$. Finally, we use a general distribution $\mathcal{D}_h$ to generate node fitness, which can be power-law, exponential or Poisson distribution ($\Pr[h(v) = k] \propto {\beta_f}^k e^{-\beta_f}/k!$), where $\beta_f \in [0, \infty)$ is the model parameter and $h(v)$ is the fitness of v. We use another parameter h_{max} to limit the upper bound of node fitness, i.e., $h(v) \in \{1, \ldots, h_{max}\}$ for each node v.

B. Generation Algorithm

To generate a graph with n nodes, we start with a single node v_1 (and an additional virtual node), and sequentially add edges to the graph. We employ a *preferential node selection procedure* to determine the two endpoints of each edge. Once the virtual node is selected as an endpoint, we add it to the graph and place another virtual node, until the graph contains n nodes. According to the implementation of the preferential node selection procedure, we propose the baseline algorithm and two efficient algorithms.

TPA-U-RW (baseline). Since the preference of each node can be computed by the attachment function, we can implement preferential node selection by the roulette wheel. Since roulette wheel selection incurs linear complexity with respect to the number of existing nodes, the algorithm needs $O(nm)$ time to generate a graph with n nodes and m edges.

TPA-U-SA. The introduction of time decay in TPA inherently prevents the application of optimization techniques for generating Barabási-Albert graphs. Hence, we adopt *logarithmic binning* to place nodes into a sequence of bins (called *T-Bucket*) according to their insertion time (see Fig. 1). To this end, preferential node selection can be decomposed into inter-bucker selection (to choose a *T-Bucket* by its weight) and the following intra-bucket selection (to choose a node inside the bucket), while both procedures can be handled efficiently. For *TPA-U-SA*, we simply implement *T-Buckets* as an array (called *D-Array*), and use stochastic acceptance algorithm for intra-bucket node selection. The time complexity is bounded by $O(md_{max}/\bar{d})$ where d_{max} (resp. $\bar{d}$) stands for maximum (resp. average) degree, and is asymptotically more efficient than the stochastic acceptance algorithm for Barabási-Albert model.

TPA-U-Hybrid. To further accelerate intra-bucket selection for large-sized *T-Buckets*, we implement *T-Buckets* as ROLL-trees [1] (called *D-Tree*) for those whose size is above a predefined threshold, as shown in Fig. 1. The total amount of data transfer from *D-Array* to *D-Tree* can be bounded, and the algorithm is practically more efficient than *TPA-U-SA*.

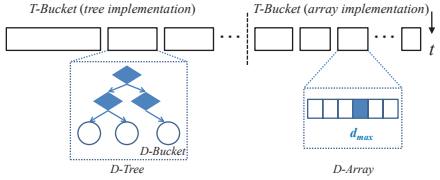

Fig. 1. Data structure for the *TPA-U-Hybrid* algorithm.

III. Experiments

We empirically evaluate the properties of synthetic graphs generated by the TPA model. The degree distribution of TPA graph follows power law with various function forms of aging and node fitness. As opposed to existing sequence-of-nodes PA models, TPA graphs also exhibit several temporal properties such as shrinking diameter. We also demonstrate that temporal preferential attachment does exist in real-world graphs. For the evaluation of generation algorithms, *TPA-U-Hybrid* can generate a TPA graph with one billion edges in about five minutes on a commodity machine.

References

[1] A. Hadian, S. Nobari, B. Minaei-Bidgoli, and Q. Qu, "Roll: Fast in-memory generation of gigantic scale-free networks," in *SIGMOD*, pp. 1829–1842, ACM, 2016.

[2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.