When Labels Fall Short: Property Graph Simulation via Blending of Network Structure and Vertex Attributes

Arun V. Sathanur
Pacific Northwest National Laboratory, Seattle, WA, USA
arun.sathanur@pnnl.gov

Sutanay Choudhury
Pacific Northwest National Laboratory, Richland, WA, USA
sutanay.choudhury@pnnl.gov

Cliff Joslyn
Pacific Northwest National Laboratory, Seattle, WA, USA
cliff.joslyn@pnnl.gov

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

ABSTRACT

Property graphs can be used to represent heterogeneous networks with labeled (attributed) vertices and edges. Given a property graph, simulating another graph with same or greater size with the same statistical properties with respect to the labels and connectivity is critical for privacy preservation and benchmarking purposes. In this work we tackle the problem of capturing the statistical dependence of the edge connectivity on the vertex labels and using the same distribution to regenerate property graphs of the same or expanded size in a scalable manner. However, accurate simulation becomes a challenge when the attributes do not completely explain the network structure. We propose the Property Graph Model (PGM) approach that uses a label augmentation strategy to mitigate the problem and preserve the vertex label and the edge connectivity distributions as well as their correlation, while also replicating the degree distribution. Our proposed algorithm is scalable with a linear complexity in the number of edges in the target graph. We illustrate the efficacy of the PGM approach in regenerating and expanding the datasets by leveraging two distinct illustrations. Our open-source implementation is available on GitHub [1].

KEYWORDS

Property Graphs, Attributed Graphs, Joint Distribution, Graph Generation, Label Augmentation, Label-Topology Correlation

1 INTRODUCTION

Most real-world datasets that naturally lend themselves to a graph representation also contain significant amounts of label (or attribute) information. This situation is promoting the popularity of property graphs: multi-graphs where the vertices and edges are labeled with key-value pairs [10]. For example, the Microsoft Academic Graph has labels such as affiliation, field of study, etc., for every person. These attributes help answer questions such as: 1) how strong are collaborations between two fields? 2) where is a person with a certain affiliation and field of study likely to publish most? Similar motivating examples are abound in other domains such as bioinformatics (protein-interaction networks), medicine (clinical records) and cyber-security (network-traffic data). The need for accurate and scalable simulation arises as an important capability for property graphs. We often need to re-generate datasets with equivalent properties for privacy reasons, or expand a dataset by multiple orders of magnitude for benchmarking studies.

The Problem We consider the problem of capturing the relationships between given and (in general) correlated vertex labels and edge connectivity in property graphs through the use of two different joint distributions. We show that a straightforward approach to capturing the label-structure relationships can be accuracy-limited when the given labels cannot explain the structure completely. We mitigate this problem by modeling the dependence of the edge connectivity on the vertex labels and the structure itself via the introduction of an augmented label that categorizes the vertex degree distribution. We demonstrate the modeling of graphs with vertices of the same type, connected by one specific type of relationship. Property graphs with heterogeneous vertices and multiple relationships can be modeled by introducing vertex types as new labels and building multiple distributions for the typed edges.

Contributions Our Property Graph Model (PGM) retains the *generative* nature of the Multiplicative Attribute Graph (MAG) model [4] by expressing the probability of edge connection as a function of the vertex labels. However, while MAG deals with latent labels, PGM caters to correlated, meaningful real-world labels. In this way it is similar to the Attribute Graph Model (AGM) approach [8]. PGM has the added benefit of not needing to assume a model for the graph topology, making it general enough to model property graphs across domains. The use of label and edge categories to define multinomial distributions provides for a scalable implementation linear in the number of edges required in the target dataset. Finally, we demonstrate our results on two datasets: a synthetic dataset generated by a role-based approach [3] and a real-world dataset extracted from the Facebook Social Graph [5].

2 THE BASIC PGM APPROACH

Consider a source property graph $G_S = \langle V_S, E_S, L, L(V_S)\rangle$ with the set of vertices V_S and the set of edges $E_S \subseteq V_S \times V_S$. $L = \{L_k\}_{k=1}^{M}$ is a set of M vertex label sets. Associated with the k^{th} label is L_k,

[1] https://github.com/propgraph/pgm

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133065

the set of possible label values for that label and $n_k = |L_k|$. For example, in a social graph, the first label, *Income-Range*, may have 6 possible values where as the second label, *Education-Level*, may have 4 possible values. Associated with each vertex $v_i \in V_S$ is a random M-vector $\bar{L}(v_i) = \left\langle l_1^i, l_2^i, \ldots, l_k^i, \ldots, l_M^i \right\rangle$ drawing a label value $l_k^i \in L_k$, for each of the M labels. We denote by $L(V_S)$, the set of all the $|V_S|$ label value vectors in one to one correspondence with the set of vertices V_S.The target property graph $G_T = \langle V_T, E_T, L, L(V_T)\rangle$ is defined analogously, and is generated by capturing appropriate statistics on G_S. Note that both G_S and G_T share the same set of vertex labels L.

Each realized vertex label vector $\bar{L}(v_i)$ can be considered as a draw from the set of joint label assignments $\mathcal{L} = \overset{M}{\underset{k=1}{\times}} L_k$. There are $N = \prod_{k=1}^{M} n_k$ possible joint label assignments called *label categories* and the j^{th} label category is denoted by c_j. In doing so, we flatten the joint distribution to a multinomial label distribution P_L over these N categories such that $p_j = P_L\left(c_j\right)$, and $\sum_{j=1}^{N} p_j = 1$. With the observations of the vertex labels in the source dataset G_S, we can estimate the parameters p_j via the maximum-likelihood method as

$$P_L\left(c_j\right) = \frac{\sum_{i=1}^{|V_S|} 1_{c_j}(\bar{L}(v_i))}{|V_S|}. \tag{1}$$

Here the indicator function is 1 only when the label vector for vertex i is equal to the joint label category c_j.

Next, we model the edge connectivity by a joint distribution over pairs of label categories $\left(c_j, c_{j'}\right)$ which we call *edge categories*. We denote *this* distribution by P_C. P_C is defined over the sample space $\mathcal{L} \times \mathcal{L}$ and has one entry per pair of label vector realizations. P_C can be estimated from data as

$$P_C\left(\left(c_j, c_{j'}\right)\right) = \frac{\sum_{\langle v_i, v_{i'}\rangle \in E_S} 1_{(c_j, c_{j'})}\left(\bar{L}(v_i), \bar{L}(v_{i'})\right)}{|E_S|}. \tag{2}$$

Here the indicator function is 1 only when the two vertices v_i and $v_{i'}$ have an edge between them and their label vectors take on categories c_j and $c_{j'}$ respectively. Note that for undirected graphs, where the order of c_j and $c_{j'}$ does not matter, P_C can be represented as a more compact multinomial distribution with $\hat{N} = \binom{N}{2}$ categories. When we draw an edge from P_C, the successful category gives the vertex label categories corresponding to the two end points that form the edge. Using a data structure such as a map (*C2V* in Algorithm 1), the participating vertices can be randomly drawn from the pools of vertices corresponding to those label categories. Drawing the edges from the multinomial distribution P_C renders the algorithm linear in the number of edges required as opposed to a naive implementation over vertex pairs which will lead to an algorithm that is quadratic in the number of vertices required. Algorithm 1 describes the basic PGM approach.

Lines 6 and 7 compute the label and edge connectivity distributions P_L and P_C from the source graph dataset G_S. Lines 9-11 sample from the distribution P_L, the vertex label set $L(V_T)$ for the target graph. Lines 14-17 construct the edge set E_T by drawing one edge at a time by sampling from a multinomial distribution over the edge categories. The resultant vertices to be connected are drawn at random from the sets of vertices corresponding to the label categories obtained from the edge category. Self and repeated edges can be removed by post-processing.

Algorithm 1 The input to the algorithm is the source property graph dataset D_S and the number of vertices and edges in the target property graph - $n_t = |V_T|$ and $m_t = |E_T|$. The output is the target property graph $\langle V_T, E_T, L, L(V_T)\rangle$.

```
1:  procedure PGM-BASIC(D_S, n_t, m_t)
2:      ⟨V_S, E_S, L, L(V_S)⟩ = processSourceDataSet (D_S)
3:      G_T = simAttrGraph(⟨V_S, E_S, L, L(V_S)⟩, n_t, m_t)
4:  end procedure
5:  procedure simAttrGraph(⟨V, E, L, L(V)⟩, n_t, m_t)
6:      P_L = computeVertexLabelDist (V, L(V))
7:      P_C = computeEdgeConnectivityDist (V, E, L(V))
8:      V_X = φ, L(V_X) = φ, E_X = φ
9:      for idx = 1 to n_t do
10:         (v, L̄(v)) = sampleFromMultiNomialDist (P_L)
11:         V_X = V_X ∪ {v}, L(V_X) = L(V_X) ∪ {L̄(v)}
12:     for i = 1 to N do        ▷ Create map C2V for all N categories
13:         C2V [c_i] = Set of all vertices with label category c_i
14:     for idx = 1 to m_t do                ▷ Draw one edge at a time
15:         [c_1, c_2] = sampleFromMultiNomialDist(P_C)
16:         Draw v_1 and v_2 at random from C2V [c_1] and C2V [c_2]
17:         E_X = E_X ∪ {(v_1, v_2)}
18:     return ⟨V_X, E_X, L, L(V_X)⟩
19: end procedure
```

3 WHEN LABELS FALL SHORT

We use two example graphs with contrasting properties to illustrate the strengths and limitations of the PGM method. The first example is a role-based graph [3] such as an enterprise network where the connectivity depends on roles that the vertices serve [2]. Thus, it is possible that there is a high chance of an edge between a SERVER-CLIENT pair while the chance of an edge between a SERVER-SERVER or a CLIENT-CLIENT pair is small. By considering two binary labels that can explain the edge connectivity, we synthesized a role-based graph with 2000 vertices and 90,000 undirected edges. Our next example consists of an anonymized Facebook social graph from the SNAP website [5]. The data is available as a number of ego-nets, each associated with a large number of binary vertex features that vary across the ego-nets. We collected the 4 labels that were common to all vertices across the ego-nets and leveraged the combined graph for our experiments. The graph has around 4000 vertices and 88,000 undirected edges with nearly a power-law degree distribution.

We then run the steps presented in Algorithm 1 to re-generate target property graphs of same size as the source property graphs. We compare the distributions P_L and P_C. The design of the algorithm ensures that P_L and P_C for the source and target distributions will match well in expectation and the same was verified. We also quantify the comparison with respect to the degree distributions between the ground truth graph and the regenerated graph by means of the Jenson-Shannon Divergence (JSD) measure [6]. JSD is small when the distributions are closer to each other.

The results for both the example datasets are shown in Figure 1. The top sub-figure shows the degree distribution comparisons between the source and regenerated versions of the role-based graph, for which there is a very good match. The degree distribution is plotted as a complementary cumulative distribution function (CCDF). The bottom sub-figure shows comparisons for the Facebook graph (on a log-log scale) for which we don't see a good match.

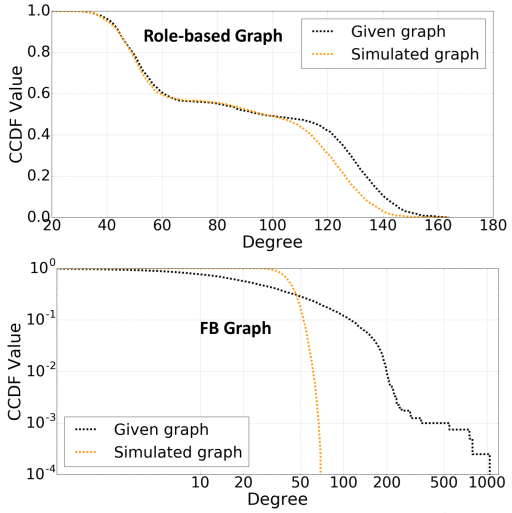

Figure 1: *Top:* Degree distribution comparison (linear scale) when graph structure is fully explained by the given labels. (JSD = 0.036). *Bottom:* Degree distribution comparison (log-log scale) for the Facebook graph where the graph structure is not fully explained by the given labels (JSD = 0.354)

The joint distribution based approach that we described in Algorithm 1 assumes that the edge connectivity is a function of label values alone.This assumption is often violated in the case of real-world datasets rendering the basic PGM approach ineffective in recovering the structural properties such as the degree distributions. It might be impossible to identify and collect all the vertex labels that can explain the graph structure. Even if all the possible labels can be collected, it is possible that the graph is grown temporally and as a result, when new vertices arrive and form edges, the connectivity is not only dependent on the label combination pairs but also on the structure of the graph itself at the time point of their arrival. The next section bridges this gap and extends the PGM approach to replicate the topological features under limited label information.

4 LABEL AUGMENTATION TO RESCUE

In [7], the authors introduce the notion that the probability of edge formation between a new vertex and a vertex already present in the graph is dependent on both the *similarity* between the two vertices and the *popularity* of already present vertex. The similarity notion refers to affinity based on vertex attributes. The popularity notion captures phenomena such as preferential attachment where vertices tend to get attached to popular vertices which are vertices with existing high degree values. Strict role-based networks such as communication networks will favor similarity while networks such as social networks will favor a combination of similarity and popularity. In the case of the PGM approach, the joint distribution implicitly encodes and generalizes the notion of similarity by quantifying the average likelihood of edge connectivity between all possible pairs of label categories (not necessarily between vertices having the same label categories). The label augmentation process that we describe next, will bring in the popularity aspect into the PGM approach.

Adopting the above philosophy, in order to better match the degree distribution of the given graph, we propose to augment the given set of labels with an additional label L_a that describe the vertex popularity. The number of values that this additional label can take on is denoted by n_a, corresponding to the division of the range of the degree values for the given graph G_S into n_a intervals. Vertex-specific label values for are assigned based on the interval in which a given vertex degree falls. We then run an iterative procedure by incrementing n_a by 1 at each step till an error measure over the source and target distributions of structural properties of interest is acceptable. In each iteration, the interval lengths can be optimally adjusted by means of an optimizer to minimize the error metric. Note that both the distributions P_L and P_C without L_a will be retained as before due to the marginalization property of the joint probability mass functions. Algorithm 2 reflects the updated flow.

Algorithm 2 The updated approach that uses label augmentation. This algorithm calls the simAttrGraph procedure in Algorithm. 1.

$\langle V_S, E_S, L, L(V_S)\rangle = processDataSet\ (D_S)$
$n_a \leftarrow 1$ ▷ n_a is the number of intervals in degree range
$error \leftarrow \infty$
procedure PGM-AUGMENTED($\langle V_S, E_S, L, L(V_S)\rangle, n_t, m_t$)
  while ($error > tolerance$) **do**
    $n_a \leftarrow n_a + 1$
    Divide degree range of the source graph into n_a intervals
    for each $v \in V_S$ **do**
      Assign $l_a(v)$ value based on the degree(v)
      Append the label vector $\bar{L}(v)$ with $l_a(v)$
    G_T = SIMATTRGRAPH($\langle V_S, E_S, L, L(V_S)\rangle, n_t, m_t$)
    $error \leftarrow computeError\ (G_S, G_T)$
end procedure

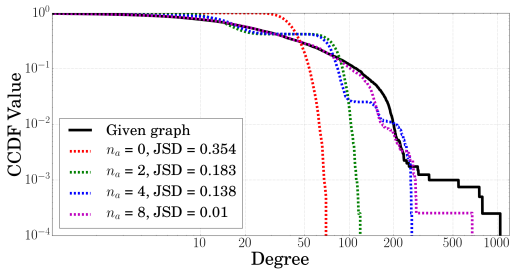

Figure 2: Degree distribution comparison (log-log) between the Facebook graph and the simulated graph with augmented label L_a and for various values of n_a.

In our experiments, for a given value of n_a, we assigned the interval lengths based on a logarithmic scale and the end-points

of the intervals were fixed. For the Facebook graph, the results of augmenting with L_a with n_a = 0,2,4,8 are shown in Figure 2. As seen the reproduction of the degree distribution is very poor without augmentation ($n_a = 0$) and gets progressively better with augmentation and by increasing n_a. The same is reflected in the observation that the JSD measure decreases with increasing n_a.

5 DATASET EXPANSION

Next we consider the expansion of the dataset by using the estimated joint distributions of the vertex labels and the edge connectivity, P_L and P_C respectively. The results are illustrated in Figure. 3 for both the role-based and the Facebook graphs. The number of vertices were expanded by 10X where as the number of edges were expanded by 12.5X. It's clear from the observed results that the PGM approach in its basic or extended form works well in expanding the attributed graphs and preserves the degree distribution shapes. Leveraging a serial implementation, we generated graphs with 1 million vertices, 31 million edges and total of 16 vertex label categories (2 binary labels and an augmented label with 4 values) in about 42 minutes on a 2.6GHz Mac workstation. Drawing of independent edges facilitates easy parallelization of the code which is ongoing.

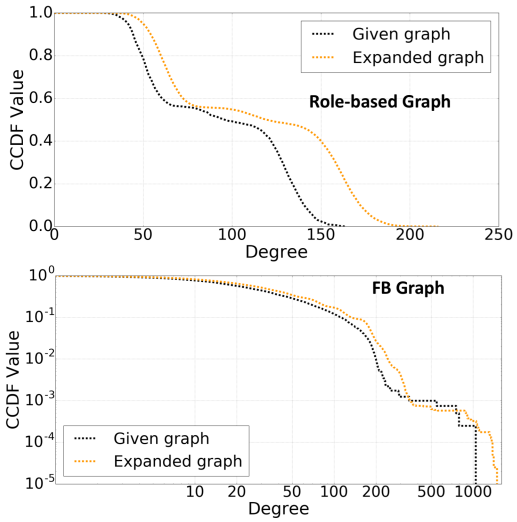

Figure 3: Comparing degree distribution shapes for 10X expansion. *Top :* Role-based graph. *Bottom :* Facebook graph with 8 label values for the augmented label (log-log scale).

6 RELATED WORK

Synthetic generation of property graphs is a nascent area of research when compared to models for network topologies. Approaches based on exponential random graph (ERG) [9] model the link probability as a linear model in a number of topological features. While such formulations are general enough to accommodate vertex labels, these methods have high computational cost beyond a few thousand vertices [8]. The Multiplicative Attribute Graph (MAG) approach models the link probability between two vertices in terms of affinities along a number of independent vertex level latent labels. However, MAG's drawback also lies in its reliance on latent labels. Generating vertex labels as observed in the data becomes difficult in a latent label based approach [8]. An alternate approach is presented in Attributed Graph Model (AGM) [8] that combines two sources of information: a) it learns the correlation between vertex labels and the graph structure, and b) exploits a known generative model for the graph topology in the form or Kronecker Product Graph Model or the Chung-Lu model. The AGM approach can perform well in replicating the graph topology and the correlation with the label values for any given set of labels. However the approach is agnostic to the explanatory power of the labels. Further, modeling and expanding arbitrary property graph datasets can be a challenge with the AGM approach that relies on specific models for the graph topology. In a recent work [1] the authors consider cloning of social networks for privacy preservation. They use the preferential attachment model to generate the graph, followed by genetic algorithms to align the statistical distribution of attributes. The use of the preferential attachment model and the optimization process limit the applicability and scalability of this approach.

7 CONCLUSIONS AND ONGOING WORK

We present a property graph generation algorithm that bridges two state of the art approaches, [4] and [8]. We initiate the simulation with observed labels and then introduce an augmented label to explain the connectivity when it is not explained by the given set of labels. Our approach reproduces or expands property graphs with a single edge relation and homogeneous vertices and is able to match the degree distributions closely. We are addressing several theoretical and implementation challenges as part of ongoing research. They include supporting heterogeneous vertices and relationships, better label augmentation strategies for large-scale dataset expansion and preservation of properties beyond degree distribution.

ACKNOWLEDGMENTS

This research was supported by the High Performance Data Analytics program at the Pacific Northwest National Laboratory (PNNL). PNNL is a multi- program national laboratory operated by Battelle Memorial Institute for the US Department of Energy under DE-AC06- 76RLO 1830.

REFERENCES

[1] A. M. Ali and *et al.* 2014. Synthetic Generators for Cloning Social Network Data. In *International Conference on Social Informatics*. Cambridge, MA.
[2] PY Chen, S. Choudhury, and A. Hero. 2016. Multi-centrality graph spectral decompositions and their application to cyber intrusion detection. In *IEEE ICASSP*.
[3] K. Henderson and *et al.* 2012. Rolx: structural role extraction & mining in large graphs. In *ACM SIGKDD*.
[4] M. Kim and J. Leskovec. 2012. Multiplicative attribute graph model of real-world networks. *Internet Mathematics* 8, 1-2 (2012).
[5] J. Leskovec and A. Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).
[6] J. Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.
[7] F. Papadopoulos and *et al.* 2012. Popularity versus similarity in growing networks. *Nature* 489, 7417 (2012), 537–540.
[8] J.J Pfeiffer III and *et al.* 2014. Attributed graph models: Modeling network structure with correlated attributes. In *WWW*. ACM.
[9] G. Robins and *et al.* 2007. An introduction to exponential random graph (p*) models for social networks. *Social networks* 29, 2 (2007).
[10] M. Simeonovski and *et al.* 2017. Who Controls the Internet?: Analyzing Global Threats Using Property Graph Traversals. In *WWW*.