PP-DBLP: Modeling and Generating Public-Private Attributed Networks with DBLP

Xin Huang*, Jiaxin Jiang*, Byron Choi*, Jianliang Xu*, Zhiwei Zhang*, Yunya Song#

*Department of Computer Science, Hong Kong Baptist University
#Department of Journalism, Hong Kong Baptist University
*{xinhuang, jxjian, bchoi, xujl, cszwzhang}@comp.hkbu.edu.hk, #yunyasong@hkbu.edu.hk

***Abstract*—In many online social networks (e.g., Facebook, Google+, Twitter, and Instagram), users prefer to hide her/his partial or all relationships, which makes such private relationships not visible to public users or even friends. This leads to a new graph model called public-private networks, where each user has her/his own perspective of the network including the private connections. Recently, public-private network analysis has attracted significant research interest in the literature. A great deal of important graph computing problems (e.g., shortest paths, centrality, PageRank, and reachability tree) has been studied. However, due to the limited data sources and privacy concerns, proposed approaches are not tested on real-world datasets, but on synthetic datasets by randomly selecting vertices as private ones. Thereto, real-world datasets of public-private networks are essential and urgently needed to such algorithms in the evaluation of efficiency and effectiveness.**

In this paper, we generate four public-private networks from real-world DBLP records, called PP-DBLP. We take published articles as public information and regard ongoing collaborations as the hidden information, which are only known by the authors. Our released datasets of PP-DBLP offer the prospects for verifying various kinds of efficient public-private analysis algorithms in a fair way. In addition, motivated by widely existing attributed graphs, we propose an advanced model of attributed public-private graphs where vertices have not only private edges but also private attributes. We also discuss open problems on attributed public-private graphs. Preliminary experimental results on our generated real-world datasets verify the effectiveness and efficiency of public-private models and state-of-the-art algorithms.

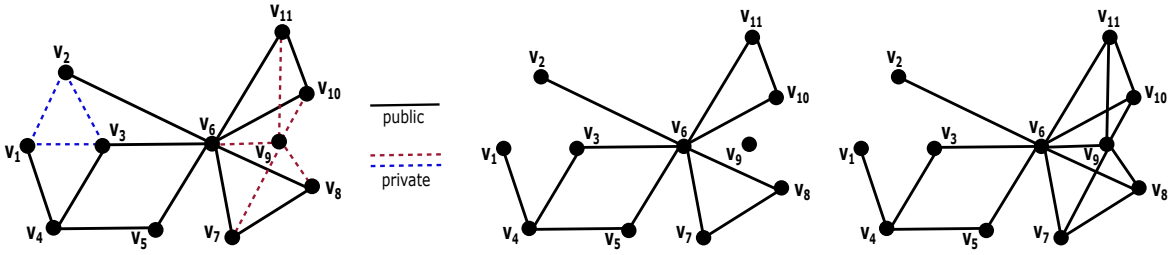
I. Introduction

Online social networks, such as Facebook, Twitter, Google+, Weibo, and Instagram, have been important platforms for the spread of information, ideas, and influence among a huge number of socially connected users. Driven by applications such as social media marketing and user behavior prediction, social network analysis, a process of investigating social structures using network and graph theories, has become a focal point of research. However, privacy issues become a major concern in the algorithmic analysis of social networks. Privacy not only affects the views of a network structure, but also controls the way information shared among social network users. As reported in a recent study [1], 52.6% of 1.4 million New York City Facebook users hid their friend's list. Such privacy protection leads to a novel graph model, called public-private graphs [2], [3], [4]. It contains a public graph, in which each vertex is also associated with a private graph. The public graph is visible to everyone, and each private graph is visible only to the corresponding user. From each users viewpoint, the social network is exactly the union of the public graph and her/his own private graph. Several sketching and sampling approaches [2] have been proposed to address essential problems of graph processing, such as the size of reachability tree [5], all-pair shortest paths [6], pairwise node similarities [7], correlation clustering [8] and so on.

In social networks, vertices usually contain attributes, e.g., a person has information including name, interests, skills, and so on. Recent study [2] focuses on one essential aspect of topological structure of public-private graphs. However, another important issue of vertex attributes has not been investigated yet. In many real-world applications, both the graph topological structure and the vertex properties are important [9]. In this paper, we model the public-private networks with vertex attributes and give a formulation of attributed public-private networks by considering the public and private vertex attributes. More importantly, as far as we know, to date there exist no publicly released datasets of real-world public-private networks. Both [2] and [3] use public graphs to simulate public-private graphs, by randomly selecting vertices and regarding their incident edges as private edges. Therefore, it is desirable to have real-world datasets of public-private networks as benchmarks for fair experimental evaluations. This work generates the real-world datasets of public-private networks from real-life DBLP records, denoted by PP-DBLP. We have also publicly released the PP-DBLP datasets to the community.[1] To summarize, this paper makes the following contributions:

- We generate and release a series of real-world public-private network datasets, according to the public and private information on a DBLP network. The released datasets offer the prospects to verify various kinds of public-private graph algorithms in a fair way. We conduct experiments on the PP-DBLP datasets to validate the efficiency of state-of-the-art algorithms (Section II).
- We formally propose a new model of attributed public-private networks, where vertices have private attributes. We highlight two promising directions on the attributed public-private networks and generate corresponding PP-DBLP datasets with attributes.(Section III).

[1] https://github.com/samjjx/pp-data



(a) Public-private graph (b) Public graph G (c) In the view of v_9: $G \cup G_{v_9}$

Figure 1. An example of an undirected and simple public-private graph. In Figure 1(a), public edges are depicted in solid black lines and private edges are depicted in dashed edges. The red edges incident to vertex v_9 are private to v_9. The blue edges incident to vertex v_3 and the edge (v_1, v_2) are private to v_3. Figure 1(b) shows the public graph G consisting of all the solid black edges. Figure 1(c) shows the structure of public-private graph in the view of v_9.

II. Public-Private Networks

In this section, we first introduce a public-private graph model for online social networks. Then, we generate real-world datasets of public-private DBLP networks (PP-DBLP), and compare state-of-the-art of public-private graph processing algorithm in efficiency on PP-DBLP datasets.

A. Public-Private Graph Model

We present the model of a public-private graph $\mathcal{G}$ [2] as follows. Given a public graph $G = (V, E)$, the vertex set V represents users, and the edge set E represents connections between users. For each vertex u in the public graph G, u has an associated private graph $G_u = (V_u, E_u)$, where $V_u \subseteq V$ are the users from public graph and the edge set E_u satisfying $E_u \cap E = \emptyset$. The public graph G is visible to everyone, and the private graph G_u is only visible to user u. Thus, in the view of user u, the user u can see the structure of graph is the union of public graph and its own private graph as $G \cup G_u = (V, E \cup E_u)$. Let $V_{private} = \{u \in V : E_u \neq \emptyset\}$ and $E_{private} = \{(v, w) \in E_u : u \in V_{private}\}$. Note that $E_{private} \cap E = \emptyset$, and each edge presented in different private graphs only counts once in the private edge set $E_{private}$ of $\mathcal{G}$.

Example 1: Consider a pubic-private graph with 11 vertices in Figure 1(a). The pubic-private graph consists of two kinds of relationships: public edges and private edges. The public edges are depicted in solid lines, e.g., (v_3, v_6), indicating that the public relationship (v_3, v_6) can be viewed by every vertex in graph G. The private edges are depicted in dash lines, e.g., (v_6, v_9). The private relationship between v_6 and v_9 can be only viewed by vertices v_6 and v_9 that are involved in the private relationship. Thus, Figure 1(b) shows the public graph G that consists of all public edges. In the view of vertex v_5, the structure of public-private graph is indeed as G in Figure 1(b), since the private graph G_{v_5} is empty. Figure 1(c) shows the structure of public-private graph in the view of v_9, which is richer than the public graph G in Figure 1(b). Because vertex v_9 can access all private relationships in private graph G_{v_9}.

B. Real-world Public-Private DBLP Networks

To date, there exists no publicly released datasets of real-world public-private networks, in light of privacy concerns. All previous algorithms for public-private social networks [2], used public social networks to simulate public-private graphs, by randomly selecting some vertices and hiding their incident edges as private edges from the public graph [2], [3]. To verify competitive algorithms in a fair way, we propose one following approach to generate real-world datasets of public-private DBLP collaboration networks (PP-DBLP), according to the public and private information on DBLP records [10].

Table I
Network statistics

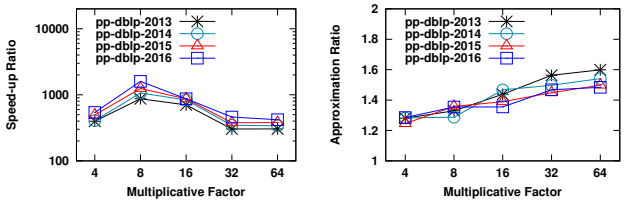
Network	$\|V\|$	$\|E\|$	$\|V_{private}\|$	$\|E_{private}\|$	$\tau(\mathcal{G})$
PP-DBLP-2013	1,791,688	5,187,025	825,170	2,636,570	0.086
PP-DBLP-2014	1,791,688	5,893,083	686,292	1,930,512	0.087
PP-DBLP-2015	1,791,688	6,605,428	515,549	1,218,167	0.087
PP-DBLP-2016	1,791,688	7,378,090	263,937	445,505	0.083

The intuition is the information of one accepted paper gets known in *public* is usually later than the co-author collaboration happened in *private*. In addition, such collaborations are always only known for authors themselves in person, and can't be aware by others. Thus, we take collaboration relationships in the published papers as public edges, and regard collaboration relationships in the ongoing works as private edges that are only known by their authors. Note that, if two authors have a collaboration relationship in public, then their private ongoing collaboration is not accounted as a private edge.

The public-private DBLP network is constructed as follows. We first obtain the DBLP raw data published in 2017 [10]. Next, we select one particular timestamp Y to distinguish the published yet papers and on-going papers. For example, taking the cut-off timestamp Y as 2013/01/01, all collaborations happened before timestamp Y are regarded as public edges and the collaborations happened on and after timestamp Y are taken as private edges. Then, we construct the public graph. We sort all papers in the increasing order of published dates in DBLP. For each paper p published before timestamp Y, we consider each author of this paper as a vertex, and add public edges between any pair of authors in this paper. Similarly, we construct all private graphs in the following. For each paper p published on and after timestamp Y, if the authors u, v do not have a public edge, we add a private edge between vertices u and v. We generate 4 PP-DBLP datasets using four timestamps of Y in {2013-01-01, 2014-01-01 2015-01-01, 2016-01-01}. The network statistics are shown in Table I.

C. Evaluations on PP-DBLP datasets

We use real-world datasets of PP-DBLP to evaluate two pubic-private graph algorithms proposed by Chierichetti et al.



(a) Efficiency Improvement (b) Quality Approximation

Figure 2. Evaluation of shortest path approximation on PP-DBLP networks

[2]: shortest path approximation and personalized PageRank. Sampling algorithms are developed to precompute the public graph G offline, and then run the online update algorithm on private graph G_u with samples of G. We use the publicly available implementation of algorithms [2] and set all the same parameters with [2] by default. We randomly select private graphs and report the running time and accuracy of each task averaged over 50 independent tests.

Shortest Path Approximation. Given a vertex u in public-private graph $G \cup G_u$, this task is to compute the shortest path from u to another arbitrary vertex in $G \cup G_u$. We evaluate the performance of one sampling algorithm of shortest path [2] with one baseline method of classical Dijkstra's algorithm [11]. Figure 2(a) and 2(b) respectively shows the results of efficiency improvement and quality approximation varied by multiplicative factor. [2] achieves the great improvement of efficiency (more than 300 times faster than the baseline method) and obtains a good balance of shortest path approximations (no greater than 1.6 times of the optimal answer) on all the multiplicative factors. Figure 2(b) shows that the approximation improves with the decreasing multiplicative factor, due to more samples used in the algorithm.

Personalized PageRank. Given a vertex u in public-private graph $G \cup G_u$, the problem of personalized PageRank (PPR) is to find node similarities to u for all vertices in $G \cup G_u$. We compare two methods. The first one is personalized PageRank using heuristic [2]. The second one is a baseline method to directly apply the algorithm of Andersen et al. [12] on graph $G \cup G_u$. The results obtained by the baseline are used as the ground-truth PPR ranking of u. Table II reports the performance of efficiency and accuracy of personalized PageRank using heuristic [2] on PP-DBLP networks. In term of efficiency comparison, the heuristic algorithm [2] is faster by three orders of magnitude than the baseline [12], which are shown in the column of speed-up ratio in Table II. Table II also shows the accuracy of the ranking computed by the heuristic algorithm w.r.t. the ground-truth PPR ranking, in terms of three measured metrics: the Root Mean Square Error (RMSE), the Cosine Similarity, and the Kendall-τ index. In terms of RMSE, the heuristic algorithm produces ranking achieving the RMSE close to 0 on all datasets; In terms of cosine similarity, it obtain nearly 1; In terms of the Kendall-τ correlation of the first 50 positions of the rankings, the score of τ@50 is still quite high falling in [0.5370, 0.6067]. Similar performance of efficiency and accuracy are also reported on other datasets in [2].

[2]https://github.com/aepasto/public-private

Table II
EFFICIENCY AND ACCURACY OF PERSONALIZED PAGERANK USING HEURISTIC [2] ON PP-DBLP NETWORKS.

Network	Speed-up Ratio	RMSE	Cosine	τ@50
PP-DBLP-2013	6646	0.0036	0.9907	0.6007
PP-DBLP-2014	5746	0.0034	0.9908	0.6067
PP-DBLP-2015	5462	0.0041	0.9906	0.5715
PP-DBLP-2016	6319	0.0041	0.9890	0.5370

III. ATTRIBUTED PUBLIC-PRIVATE NETWORKS

The proliferation of rich information available for DBLP records, e.g., keywords of titles in papers. This gives rise to an attributed graph where graph vertices are associated with a number of attributes. In this section, we first define an advanced graph model of attributed public-private networks. Then, we extend the approach of generating PP-DBLP to produce real-world datasets of attributed public-private networks using title keywords. Finally, we offer open problems and promising directions on attributed public-private graphs.

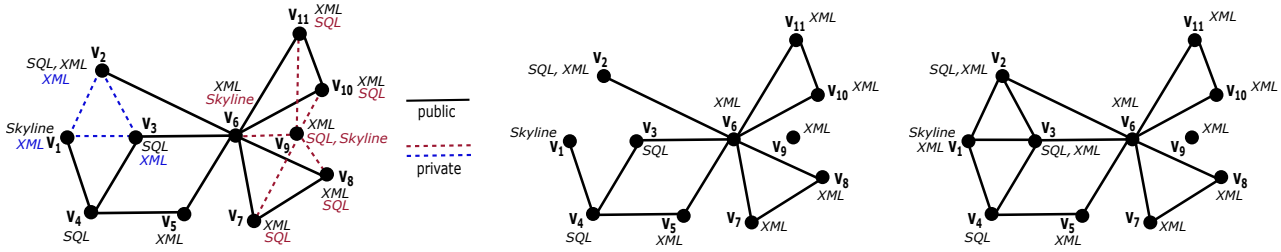
A. Attributed Public-Private Graph Model

An attributed public-private graph of $\mathcal{G}$ is modeled as follows. Given an attributed public graph $G = (V, E, A)$, where the vertex set V representing users, the edge set E representing connections between users, and the public attribute set $A(u)$ describes the public attributes of a user $u \in V$. For each user u in the public graph, u has an attributed private graph $G_u = (V_u, E_u, A_u)$, where $V_u \subseteq V$ is a set of users from the public graph, private edge set $E_u \cap E = \emptyset$, and $A_u(v)$ represents the private attributes of vertex $v \in V_u$ that are visible to u. The public attributed graph G is visible to everyone, and the private attributed graph G_u is only visible to user u. In terms of network structure, attributed public-private graphs have no difference with the public-private graphs in Section II. In terms of attributes, consider the attributed public-private graphs $G \cup G_u$, each vertex u can access both public and private attributes of vertex v, i.e., $A_v \cup A_u(v)$.

Example 2: Figure 3 (a) shows an example of attributed public-private graph, which has the same graph structure of Figure 1 (a). Public attributes are in black, and private attributes are in blue and red. Consider the vertex v_3. The public attributes of v_3 is $A(v_3) = \{`SQL'\}$. Attributes in blue (e.g., the attribute of 'XML' associated with vertices v_1, v_2 and v_3) are private and visible to v_3. Thus, $A_{v_3}(v_3) = A_{v_3}(v_1) = A_{v_3}(v_2) = \{'XML'\}$. Attributes in red (e.g., vertex v_6's attribute of 'Skyline') are private and visible to v_3. Figure 3(b) shows the public attributed graph G consisting of all public edges and public attributes that are visible to everyone. Figure 3(c) shows the attributed graph $G \cup G_{v_3}$ in the view of v_3. The attributes of v_1 is $\{`Skyline', `XML'\}$ as a result of the union of public attributes and private attributes, i.e., $A(v_1) \cup A_{v_3}(v_1)$, showing that v_1 extends her/his research interests.

B. Attributed Public-Private DBLP Networks

To construct the attributed public-private DBLP networks, we add attributes into vertices on PP-DBLP in Section II-B as



(a) Attributed public-private graph (b) Attributed public graph G (c) In the view of v_3: $G \cup G_{v_3}$

Figure 3. An example of attributed public-private graph. In Figure 3(a), public attributes are in black. Private attributes are in blue and red, which respectively are visible to v_3 and v_9. Figure 3(b) shows the attributed public graph G consisting of all public edges and public attributes. Figure 3(c) shows the attributed graph $G \cup G_{v_3}$ in the view of v_3.

follows. For each author, we collect keywords in the title of all published articles and extract the most frequent keywords as the public attributes. For the private attributes, let's consider one author u and its attributed private graphs G_u. For each author v in G_u, the private attributes of v as $A_u(v)$ are the most frequent keywords from the title of all ongoing papers involving authors v and u. To select representative keywords, we set each number of public attributes and private attributes is no greater than a maximum threshold of 5, i.e., $|A(v)| \leq 5$ and $|A_u(v)| \leq 5$. The difference of public attributes A_v and private attributes $A_u(v)$ shows the changed research interests of author v. Note that, $A_v \cap A_u(v) \neq \emptyset$ may hold. We use $\theta_u(v)$ to quantify the overlapping ratio of public attributes and private attributes of vertex v in graph G_u, denoted by $\theta_u(v) = \frac{|A_v \cap A_u(v)|}{|A_v \cup A_u(v)|}$. And, the $\tau(u) = \frac{\sum_{v \in V_u} \theta_u(v)}{|V_u|}$ represents the average ratio of overlapping attributes among the authors in G_u. For an attributed public-private graph $\mathcal{G}$, we propose $\tau(\mathcal{G})$ to measures the ratio of overlapping public-private attributes for all private graphs, denoted by $\tau(\mathcal{G}) = \frac{\sum_{u \in V_{private}} \tau(u)}{|V_{private}|}$. Table I reports the statistic $\tau(G)$ for all PP-DBLP datasets.

C. Open Problems

We offers two open problems of keyword search and community search in attributed public-private networks as follows.

- **Keyword search in attributed public-private networks.** Keyword search finds users in the vicinity of a given user with similar keywords [13], [14], [15]. Keyword search queries in an attributed public-private network are generated from a vertex that looks for nearest vertices with certain keywords, w.r.t., private structure and attributes.
- **Community search in attributed public-private networks.** Attributed community search aims at finding the densely-connected subgraphs containing given query nodes with similar attributes [16]. In an attributed public-private graph, given a community search query, the query asked by a user u needs to be considered in graph $G \cup G_u$, w.r.t., private structures and attributes.

IV. Conclusions and Discussions

In this paper, we develop a new model of attributed public-private networks w.r.t. the information of vertices in many real-world networks. In addition, we provide real-world PP-DBLP datasets for attributed public-private networks, which are useful to further research of public-private graphs. Besides PP-DBLP, our future plan is to build a real-world pubic-private Facebook social network by conducting a survey of Facebook users, who will be asked to manually identify all of private relationships that they hid. The estimated number of such interviewed users is around 100.

Acknowledgment

This work is supported by the Hong Kong General Research Fund (GRF) Project Nos. HKBU 12200917, 12232716, 12258116 and National Natural Science Foundation of China (NSFC) Project Nos. 61702435, 61602395.

References

[1] R. Dey, Z. Jelveh, and K. Ross, "Facebook users have become much more private: A large-scale study," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, 2012, pp. 346–352.

[2] F. Chierichetti, A. Epasto, R. Kumar, S. Lattanzi, and V. Mirrokni, "Efficient algorithms for public-private social networks," in *KDD*, 2015, pp. 139–148.

[3] A. Archer, S. Lattanzi, P. Likarish, and S. Vassilvitskii, "Indexing public-private graphs," in *WWW*, 2017, pp. 1461–1470.

[4] B. Mirzasoleiman, M. Zadimoghaddam, and A. Karbasi, "Fast distributed submodular cover: Public-private data summarization," in *Advances in Neural Information Processing Systems*, 2016, pp. 3594–3602.

[5] E. Cohen and H. Kaplan, "Summarizing data using bottom-k sketches," in *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*. ACM, 2007, pp. 225–234.

[6] A. Das Sarma, S. Gollapudi, M. Najork, and R. Panigrahy, "A sketch-based distance oracle for web-scale graphs," in *WSDM*, 2010, pp. 401–410.

[7] T. H. Haveliwala, "Topic-sensitive pagerank," in *WWW*, 2002, pp. 517–526.

[8] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[9] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu, "Clustering large attributed information networks: an efficient incremental computing approach," *DMKD*, vol. 25, no. 3, pp. 450–477, 2012.

[10] *http://dblp.uni-trier.de/xml*.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2009.

[12] R. Andersen, F. Chung, and K. Lang, "Local partitioning for directed graphs using pagerank," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2007, pp. 166–178.

[13] M. Qiao, L. Qin, H. Cheng, J. X. Yu, and W. Tian, "Top-k nearest keyword search on large graphs," *Proceedings of the VLDB Endowment*, vol. 6, no. 10, pp. 901–912, 2013.

[14] Q. Zhu, H. Cheng, and X. Huang, "I/o-efficient algorithms for top-k nearest keyword search in massive graphs," *The VLDB Journal*, vol. 26, no. 4, pp. 563–583, 2017.

[15] J. Jiang, B. Choi, J. Xu, and S. S. Bhowmick, "A generic ontology framework for indexing keyword search on massive graphs," *https://www.comp.hkbu.edu.hk/∼jxjian/tr2018.pdf*, 2018.

[16] X. Huang and L. V. Lakshmanan, "Attribute-driven community search," *Proceedings of the VLDB Endowment*, vol. 10, no. 9, pp. 949–960, 2017.