2020 IEEE 36th International Conference on Data Engineering (ICDE)

Efficient Structural Clustering in Large Uncertain Graphs

Yongjiang Liang
Department of Computer Science
Florida State University
Tallahassee, USA
liang@cs.fsu.edu

Tingting Hu
Department of Mathematics
Embry-Riddle Aeronautical University
Daytona Beach, USA
hut@erau.edu

Peixiang Zhao
Department of Computer Science
Florida State University
Tallahassee, USA
zhao@cs.fsu.edu

***Abstract*—Clustering uncertain graphs based on the probabilistic graph model has sparked extensive research and widely varying applications. Existing structural clustering methods rely heavily on the computation of pairwise reliable structural similarity between vertices, which has proven to be extremely costly, especially in large uncertain graphs. In this paper, we develop a new, *decomposition-based* method, ProbSCAN, for efficient reliable structural similarity computation with theoretically improved complexity. We further design a cost-effective index structure UCNO-Index, and a series of powerful pruning strategies to expedite reliable structural similarity computation in uncertain graphs. Experimental studies on eight real-world uncertain graphs demonstrate the effectiveness of our proposed solutions, which achieves orders of magnitude improvement of clustering efficiency, compared with the state-of-the-art structural clustering methods in large uncertain graphs.**

***Index Terms*—Uncertain Graphs, Structural Clustering**

I. Introduction

In the modern networked world, graphs have been widely used for modeling and interpreting interconnected relationships between network entities. However, real-life networks are oftentimes associated with *uncertainty* caused by the noise, distortions, and measurement errors arising in every stage of the graph computation pipeline [1]. As a result, extensive research on real networks has shifted focus onto *uncertain graphs*, where each edge is associated with a probability indicating the likelihood of the existence of that edge in the networks [2]. In this paper, we consider a fundamental graph analytical operation, *structural clustering*, which has found widely varying applications in real-world uncertain graphs. The objective of structural clustering is to partition an uncertain graph based solely on its interlinked topological structures, such that nodes within the same cluster are closely connected, while those belonging to different clusters are far apart in a *probabilistic* sense [3].

An existing solution for structural clustering in uncertain graphs, referred to as USCAN [3], relies primarily on the key notion of *reliable structural similarity*, which quantifies the probability of the event that two vertices are structurally similar in the uncertain graph, in terms of two parameters ϵ and η. The notion of *reliable core* can be further defined to identify the nodes that have a sufficient number of reliable structure-similar neighboring nodes, quantified by the parameter μ. These reliable cores, once identified from the uncertain graph, can uniquely determine a cluster. However, USCAN suffers from severe performance and scalability issues, due in particular to the sheer cost of reliable structural similarity computation, especially in real-world large uncertain graphs.

In this paper, we propose a new, and more efficient method for reliable structural similarity computation, denoted by ProbSCAN (in Section III). Inspired by [4], ProbSCAN decomposes the costly reliable structure similarity computation into smaller sub-portions, each of which is with only one variable, and thus can be computed separately and more efficiently. Specifically, given a node pair (u, v) from the uncertain graph, we prove that the time complexity of reliable structural similarity computation can be improved from $\mathrm{O}(d_m^2 \cdot \min\{k_u, k_v\})$ (in USCAN) to $\mathrm{O}((\min\{k_u, k_v\})^2)$ (in ProbSCAN), where k_u (resp. k_v) is the degree of the node u (resp. v), and d_m is the maximum node-degree of the uncertain graph. It is worth mentioning that the improved complexity is only determined by localized node-degrees of u and v, for which the reliable structure similarity is to be computed, but becomes irrelevant to the global variable, d_m. This improvement leads to an immediate and significant performance gain for structural clustering, especially in large-scale uncertain graphs.

We further develop an index structure, UCNO-Index, and a series of index-based pruning algorithms, to facilitate the computation of reliable structural similarity in uncertain graphs (in Section IV). Specifically, the size of UCNO-Index is well bounded by $O(m)$, the size of the uncertain graph. To the best of our knowledge, UCNO-Index is the first index-based solution for the structural clustering problem in uncertain graphs.

We perform extensive experimental studies on eight real-world uncertain graphs, and compare our solutions with the state-of-the-art method, USCAN [3] (in Section V). The experimental results demonstrate that our methods have achieved several orders of magnitude improvement, in terms of clustering efficiency, than USCAN.

II. Preliminaries

Given an *uncertain graph* $\mathcal{G}(V, E, p)$, where V is a set of vertices, E is a set of edges, and p is a probabilistic function $p : E \rightarrow [0, 1]$ that assigns for each edge $e \in E$ a probability value p_e. Such an edge probability is assumed to be independent of those of any other edges of E, and we consider the well-accepted *possible-world* semantics in this

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00215

paper for uncertain graph modeling [1]. Specifically, $G \subseteq \mathcal{G}$ denotes that $G(V, E_G)$ is a possible world of $\mathcal{G}$. Given a vertex $u \in V_G$ in the possible world G of $\mathcal{G}$, the *neighbors* of u in G, denoted as $N_G[u] = \{v \in V_G | (u, v) \in E_G\} \cup \{u\}$, consisting of all the adjacent vertices of u in G, including u itself.

Definition 1. [Probability of Structural Similarity] *Given an uncertain graph $\mathcal{G}$, an edge $e = (u, v) \in E$, and a similarity threshold $\epsilon \in (0, 1]$, the probability of structural similarity of e* s.t. *$\sigma(e) \geq \epsilon$, denoted by $\Pr[e, \epsilon]$, is defined as the sum of the probabilities of all possible worlds $G \subseteq \mathcal{G}$, such that the structural similarity between u and v is no less than ϵ in each possible world G:*

$$\Pr[e, \epsilon] = \sum_{G \subseteq \mathcal{G}} \Pr[G : \sigma_G(u, v) \geq \epsilon] \tag{1}$$

where $\sigma_G(u, v) = \frac{|N_G[u] \cap N_G[v]|}{|N_G[u] \cup N_G[v]|}$. □

Definition 2. [Reliable Structural Similarity] *Given an edge $e = (u, v)$ and a probability threshold η, u is* reliably structural similar *to v if $\Pr[e, \epsilon] \geq \eta$.* □

Definition 3. [(ϵ, η)-Reliable Neighbor] *The (ϵ, η)-reliable neighbors of a vertex u, denoted as $N_{(\epsilon,\eta)}[u]$, is a subset of $N_{\mathcal{G}}[u]$, in which every vertex v is reliably structural similar to u; that is, $N_{(\epsilon,\eta)}[u] = \{v \in N_{\mathcal{G}}[u] | \Pr[e = (u, v), \epsilon] \geq \eta\}$.* □

Note (ϵ, η)-reliable neighbors of a given vertex u include u itself. Intuitively, when the number of (ϵ, η)-reliable neighbors of u is large, u tends to be critical in structural clustering, and we refer to u as a *reliable core*, defined as follows,

Definition 4. [(ϵ, η, μ)-Reliable Core] *Given a similarity threshold $\epsilon \in (0, 1]$, a probability threshold $\eta \in (0, 1]$, and an integer $\mu \geq 2$, a vertex u is a (ϵ, η, μ)-reliable core if $|N_{(\epsilon,\eta)}[u]| \geq \mu$.* □

Accordingly, a vertex is a *non-core* if it is not a reliable core. Detecting reliable cores turns out to be crucial, as clusters can be identified by expanding reliable cores.

Definition 5. [Reliably Structure-reachable] *Given parameters $\epsilon \in (0, 1]$, $\eta \in (0, 1]$, and $\mu \geq 2$, a vertex v is reliably structure-reachable from u if there is a sequence of vertices $v_1, v_2, \cdots, v_l \in V (l \geq 2)$, s.t. (i) $v_1 = u, v_l = v$; (ii) for all $1 \leq i \leq l-1$, v_i is a reliable core, and $v_{i+1} \in N_{(\epsilon,\eta)}[v_i]$.* □

We formulate the structural clustering problem in uncertain graphs as follows,

Definition 6. [Structural Clustering] *Given an uncertain graph $\mathcal{G}$, and parameters $\epsilon \in (0, 1]$, $\eta \in (0, 1]$, $\mu \geq 2$, we consider computing the set $\mathcal{C}$ of reliable clusters from $\mathcal{G}$. Each* reliable cluster *$C \in \mathcal{C}$ should contain at least two vertices (i.e., $|C| \geq 2$) such that:*

- [Maximality] *For each reliable core $u \in C$, all vertices reliably structure-reliable from u must belong to C;*
- [*Connectivity] For any two vertices $v_1, v_2 \in C$, there exists a vertex $u \in C$ such that both v_1 and v_2 are reliably structure-reachable from u.* □

III. ProbSCAN

The major cost in the existing solution, USCAN [3], is to compute the reliable structural similarity ($\Pr[e, \epsilon]$) between node pairs of $\mathcal{G}$. To lower the computational complexity of $\Pr[e, \epsilon]$, we introduce a new method, named ProbSCAN. The key idea is to break down the computation of $\Pr[e, \epsilon]$ into smaller portions with fewer variables, such that each portion can be computed independently, either sequentially or in parallel, like the decomposition approach proposed in [4].

A. $\Pr[e, \epsilon]$-decomposed Computation

Proposition 1 below provides the theoretical foundation for decomposed computation of $\Pr[e, \epsilon]$.

Proposition 1. *Given an uncertain graph $\mathcal{G} = (V, E, p)$, an edge $e = (u, v) \in \mathcal{G}$, and a similarity threshold ϵ, $\Pr[\sigma(u, v) \geq \epsilon]$ can be computed as*

$$\Pr[\sigma(u, v) \geq \epsilon] = \sum_{i=0}^{k_{(u,v)}} \Pr[\sup_{\mathcal{G}}(u, v) = i] \times \left(\sum_{j=i+1}^{k_u} \Pr[d_u = j] \times \sum_{k=i+1}^{\min(k_v,\, i+\lfloor \frac{i+2}{\epsilon} \rfloor - j)} \Pr[d_v = k] \right) \tag{2}$$

where k_u is the vertex-degree of u in $\mathcal{G}$, $k_{(u,v)} = |N_{\mathcal{G}}[u] \cap N_{\mathcal{G}}[v]| - 2$ is the maximum possible support of the edge (u, v) in $\mathcal{G}$, $\sup_{\mathcal{G}}(u, v) \in [0, k_{(u,v)}]$, and $d_u \in [1, k_u]$. □

With Proposition 1, the computation of reliably structural similarity $\Pr[\sigma(u, v) \geq \epsilon]$ can be decomposed into three sub-problems: (1) computing $\Pr[\sup_{\mathcal{G}}(u, v) = i]$, (2) computing $\Pr[d_u = j]$, and (3) computing $\Pr[d_v = k]$. The sub-problems (2) and (3) are essentially the same, we thus focus on the sub-problems (1) and (2). Fortunately, both have been addressed in the previous studies [5], [6].

B. Implementation

We pre-compute $\Pr[d_v = k]$ for all vertices v, and store the values into a two-dimensional array *X*: $X[v][k] = \Pr[d_v = k], k \in [0, k_v]$. We further maintain another two-dimensional array *Y* to store the values of $\Pr[0 \leq d_v \leq k]$, i.e., $Y[v][k] = \sum_{i=0}^{k} X[v][k], k \in [0, k_v]$. This way, it takes only constant time to obtain the value $\sum_{k=i+1}^{i+\lfloor \frac{i+2}{\epsilon} \rfloor - j} \Pr[d_v = k]$, which corresponds to $Y[v][i + \lfloor \frac{i+2}{\epsilon} \rfloor - j] - Y[v][i]$.

Furthermore, while we compute $\Pr[e, \epsilon]$ in Equation 2, we always start with the vertex u with a smaller degree, and choose another vertex v with a larger degree. This way, we can reduce the computation in the second loop for d_u.

Time complexity. The time complexity to process an edge (u, v) of $\mathcal{G}$ is bounded by $O((\min\{k_u, k_v\})^2)$. As a consequence, the total cost of computing $\Pr[e, \epsilon]$ for all $e \in E$ can be bounded by $O(\sum_{(u,v) \in E} (\min\{k_u, k_v\})^2) = O(d_m \times \alpha \times m)$, where α is the arboricity of the graph $\mathcal{G}$.

Space complexity. We need to maintain two two-dimensional arrays *X* and *Y*, which consumes $O(\sum_{v \in V} k_v) = O(m)$ space. The whole graph itself consumes $O(m)$ space. Therefore, the total space complexity is bounded by $O(m)$.

IV. INDEX-BASED APPROACH

In this section, we propose a novel index structure, denoted as UCNO-Index (**u**ncertain **c**ore-**n**eighbor η-**o**rder index). The idea of UCNO-Index is to pre-select a small set S of ϵ values. We then index the reliable structural similarity of node-pairs only for the ϵ values in S. Given this index, we can efficiently obtain tight lower- and upper-bounds for reliable structural similarity of node-pairs for any ϵ, and the space cost of UCNO-Index can be well bounded by $O(m)$.

A. Index Construction

We start with the computation of all reliable cores for any given probability threshold η under specific μ and ϵ. Note that because the parameter μ cannot be larger than the maximum vertex degree, we can obtain a set of candidate reliable cores, denoted as $\{u \in V | k_u \geq \mu\}$. Next, for each specific μ and a fixed similarity threshold $\epsilon \in S$, we compute the reliable cores by the probability threshold η. Recall that a vertex u is a reliable core if the number of (ϵ, η)-reliable neighbors is at least μ. We thus have the following theorem:

Theorem 1. *Given an uncertain graph $\mathcal{G}$, a fixed pair $(\mu \geq 2, \epsilon)$, and two probability thresholds $0 < \eta \leq \eta' \leq 1$, a vertex u is a reliable core of a cluster determined by η, if it is a reliable core of a cluster determined by η'.* □

According to the monotonicity property in Theorem 1, we only need to maintain the largest value of η for each vertex u that is a reliable core given each specific parameter pair (μ, ϵ). We call such a value *core-η-threshold*, defined as follows:

Definition 7. [CORE-η-THRESHOLD] *Given an uncertain graph $\mathcal{G}$, a fix parameter pair (μ, ϵ), the core-η-threshold of a vertex u, denote as $\mathcal{CET}_u[\epsilon][\mu]$, is the largest η such that u is a (ϵ, η, μ)-reliable core determined by ϵ, μ, and η.* □

Theorem 2. *Given an uncertain graph $\mathcal{G}$, parameters μ and ϵ, the core-η-threshold of a vertex u $(k_u \geq \mu)$ is the μ-th largest value in the probabilities of structural similarity between u and its neighborhood.* □

Based on Theorem 2, for each vertex u, we compute the probabilities of structural similarity between u and its neighbors $v \in N[u]$, and sort the neighborhood in a non-increasing order based on their probabilities of structural similarity, as follows,

Definition 8. [NEIGHBOR-η-ORDER] *Given an uncertain graph $\mathcal{G}$, a vertex u and a fixed ϵ, the neighbor-η-order of u, denoted by $\mathcal{NEO}_u[\epsilon]$, is a probabilistic order of neighboring vertices such that: (i) the i-th value in $\mathcal{NEO}_u[\epsilon]$ is $(v, \Pr[(u, v), \epsilon])$, where $v \in N[u], \mathcal{CET}_u[\epsilon][\mu] = \Pr[e, \epsilon]$; and (ii) for any two vertices v_1 and v_2, v_1 occurs before v_2 if and only if $\Pr[(u, v_1), \epsilon] \geq \Pr[(u, v_2), \epsilon]$.* □

For each $\epsilon \in S$, we store the corresponding neighbor-η-order for all the vertices into UCNO-Index. The number of entries in $\mathcal{NEO}_u[\epsilon_i]$ is k_u for each vertex u and $\epsilon \in S$, and thus the size of all neighbor-η-orders can be well bounded.

TABLE I
NETWORK STATISTICS

Datasets	$\|V\|$	$\|E\|$	d_m	d	$\bar{p}$
Krogan	2559	7031	141	5.49	0.6799
DBLP	636,751	2,366,461	446	7.43	0.4487
Amazon	334,863	925,872	549	5.53	0.5001
Youtube	1,134,890	2,987,624	28,754	5.27	0.5001
Google	875,713	5,105,039	6,332	9.87	0.5001
Cit	3,774,768	16,518,948	793	6.13	0.5001
LiveJournal	3,997,962	34,681,189	14,815	17.35	0.5001
Orkut	3,072,441	117,185,083	33,313	76.28	0.5001

Theorem 3. *Given ϵ, the space cost of neighbor-η-orders for all vertices is $\sum_{u \in V, \epsilon \in S} \mathcal{NEO}_u[\epsilon] = O(m)$. Therefore, the overall space cost of the index structure,* UCNO-Index*, is $O(m)$.* □

B. Query Processing

Based on UCNO-Index, we can reduce the computational cost of $\Pr[e, \epsilon]$, by devising novel lower- and upper-bounds for $\Pr[e, \epsilon]$, given any ϵ. Specifically, given a query with parameters $\epsilon \in (0, 1], \eta \in (0, 1]$ and $\mu \geq 2$, we can categorize it in the following three cases:

1) There exists an $\epsilon' \in S$ that equals ϵ. We can answer this query in $O(m)$ time.
2) There exists an $\epsilon' < \epsilon$. We find in this case a tight upper-bound of $\Pr[e, \epsilon]$ for all the edges.
3) There exists an $\epsilon' > \epsilon$. We find in this case a tight lower-bound of $\Pr[e, \epsilon]$ for all edges.

According to Case (2), we can obtain the candidate reliable neighbors for the vertex u, which is $\{v | \Pr[e, \epsilon] \geq \eta, (v, \Pr[e, \epsilon]) \in \mathcal{NEO}_u[\epsilon]\}$. Since the vertices of $\mathcal{NEO}_u[\epsilon_i]$ are sorted in a non-increasing order of probabilities of structural similarity, we perform a binary search with the value η upon $\mathcal{NEO}_u[\epsilon]$ in order to efficiently obtain the candidate reliable neighbors of u.

V. EXPERIMENTS

We report our experimental studies in eight real-world probabilistic networks, and the detailed statistics of these networks are presented in Table I. Specifically, the average degree ($\bar{d}$) and the average probability ($\bar{p}$) are listed in the last two columns. Edge probabilities of the first two networks stem from real-world application domains, while probabilities in other network datasets are randomly assigned. We compare our solutions, ProbSCAN and UCNO-Query, with the state-of-the-art method, USCAN [3] on all eight networks. All the algorithms are implemented in C++ and compiled with g++ 7.4.0. All the experiments are performed on a Linux server running Ubuntu 18.04 with two Intel 2.3GHz ten-core CPUs and 256GB memory.

Clustering Performance. The runtime for all structural clustering algorithms under the default parameter setting, $\eta = 0.5, \epsilon = 0.5$ and $\mu = 5$, on all datasets is illustrated in Figure 1. We recognize from the experimental results that UCNO-Query is more efficient than ProbSCAN, and it is

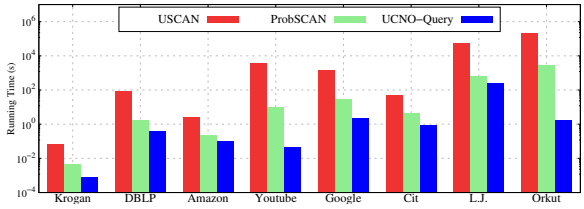

Fig. 1. Clustering Performance in Different Networks

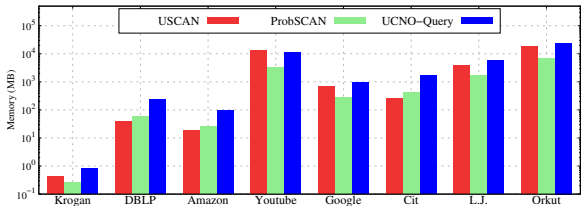

Fig. 2. Memory Consumption in Different Networks

one or two orders of magnitude faster than USCAN on all datasets. For instance, in the Krogan network, UCNO-Query only needs 0.8ms for structural clustering, while ProbSCAN and USCAN need 5ms and 70ms, respectively. In another dataset Orkut with over 100 millions edges, it takes UCNO-Query less than two seconds for structural clustering, while ProbSCAN and USCAN spend approximately $2,700$ seconds and around 2 days, respectively, in this large uncertain graph.

Space Cost. The memory consumption results of different structural clustering methods are reported in Figure 2 for different networks. In general, the memory usage of all algorithms grows proportionally when the network size, in terms of the number of edges, grows, with one exception for USCAN on the Youtube network. We note that the space cost of USCAN in Youtube is about three times of that for the Live Journal network. The reason is that the dominating factor of the space cost in this case turns out to be the maximum vertex degree d_m, rather than the network size, in this network. Additionally, the total memory consumption of UCNO-Query can always be bounded by 3.5X the size in ProbSCAN.

VI. Related Work

Uncertain Graphs. There have been a lot of fundamental querying and mining problems that have been studied in the setting of uncertain graphs, including, but not limited to, cohesive graphs detection [5], [6], reliability search [7], pattern matching [2], kNN search [8], and frequent pattern mining [9]. Bonchi *et al.* [5] study the core decomposition problem on uncertain graphs. Huang *et al.* [6] propose the concepts of local and global (k, γ)-truss that enable truss decomposition for probabilistic graphs. Jin *et al.* [7] study the distance-constraint reachability query problem in uncertain graphs. Lian *et al.* [2] propose a framework to efficiently answer RDF queries over probabilistic RDF graphs. Potamias *et al.* [8] study the problem of k-nearest neighbor search on uncertain graphs. Zou *et al.* [9] examine the problem of discovering frequent subgraph patterns on uncertain graph databases.

Structural Graph Clustering. In deterministic graphs, there have been numerous structural graph clustering methods. Xu *et al.* [10] propose the algorithm, SCAN, that can help identify densely connected graph clusters as well as hubs and outliers. The main issue of SCAN is that it has to consider all the adjacent vertex-pairs for structural similarity computation. To address this issue, Chang *et al.* [11] propose the PSCAN algorithm that identifies core vertices first. Dong *et al.* [12] develop an index-based solution, which is the state-of-the-art method for structural clustering in deterministic graphs. However, all these SCAN-based algorithms cannot be directly applied in uncertain graphs. [3] first explores the SCAN framework in the uncertain graph setting based on a new concept: reliable structural similarity, which quantifies the probability of the event that two vertices are structurally similar in a probabilistic sense in an uncertain graph.

VII. Conclusion

In this paper, we study the structural clustering problem in uncertain graphs. We develop a new, decomposition-based method, ProbSCAN, for efficient reliable structural similarity computation with theoretically improved complexity. We further design a cost-effective index structure, UCNO-Index, and powerful pruning strategies to further expedite the reliable structural similarity computation in large uncertain graphs. Experimental studies on eight real-world uncertain graphs demonstrate that our proposed methods have significantly outperformed the state-of-the-art structural clustering solutions on large uncertain graphs.

Acknowledgment

This work was supported by the National Science Foundation (Grant No. 1743142), the Air Force Office of Scientific Research (Award No. FA95501810106), and the Army Research Office (Award No. W911NF1810395). Any opinions, findings, and conclusions in this paper are those of the author(s) and do not necessarily reflect the funding agencies.

References

[1] A. Khan and L. Chen, "On uncertain graphs modeling and queries," *Proc. VLDB Endow.*, vol. 8, no. 12, pp. 2042–2043, 2015.
[2] X. Lian and L. Chen, "Efficient query answering in probabilistic rdf graphs," in *SIGMOD*, 2011.
[3] Y. Qiu, R. Li, J. Li, S. Qiao, G. Wang, J. X. Yu, and R. Mao, "Efficient structural clustering on probabilistic graphs," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
[4] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.
[5] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich, "Core decomposition of uncertain graphs," in *KDD*, 2014.
[6] X. Huang, W. Lu, and L. V. Lakshmanan, "Truss decomposition of probabilistic graphs: Semantics and algorithms," in *SIGMOD*, 2016.
[7] R. Jin, L. Liu, B. Ding, and H. Wang, "Distance-constraint reachability computation in uncertain graphs," in *PVLDB*, 2011.
[8] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, "K-nearest neighbors in uncertain graphs," in *PVLDB*, 2010.
[9] Z. Zou, H. Gao, and J. Li, "Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics," in *KDD*, 2010.
[10] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: A structural clustering algorithm for networks," in *KDD*, 2007.
[11] L. Chang, W. Li, X. Lin, L. Qin, and W. Zhang, "pscan: Fast and exact structural graph clustering," in *ICDE*, 2016.
[12] D. Wen, L. Qin, Y. Zhang, L. Chang, and X. Lin, "Efficient structural graph clustering: An index-based approach," in *VLDB*, 2017.