# Community Search for Large Profiled Graphs

kevin

## ABSTRACT

abstract


*Proceedings of the VLDB Endowment,* Vol. 9, No. 12

# 1. THE PCQ PROBLEM

DEFINITION 1 ($k$-CORE [5, 1]). *Given an integer $k$ ($k \geq 0$), the $k$-core of $G$, denoted by $H_k$, is the largest subgraph of $G$, such that $\forall v \in H_k$, $deg_{H_k}(v) \geq k$.*

DEFINITION 2 (PROFILE-TREE). *xxxx*

DEFINITION 3 (INDUCED ROOTED SUBTREE). *Given two P-trees $T$ and $S$ rooted at $r_t$ and $r_s$. $T$ is the induced rooted subtree of $S$ iff there exist a one-to-one mapping $\varphi$: $V_s \rightarrow V_t$ and $l$ denotes the label of tree node $x$, three constraints hold:*

- $\varphi(r_s) = r_t$;
- $\forall x \in V_s$, $l(x) = l(\varphi(x))$;
- $\forall (x, y) \in E_s$, $(\varphi(x), \varphi(y)) \in E_t$.

*Inuduced rooted subtree* preserves the parent-child relationships as well as corresponding labels. In addition, the root node of the trees must be preserved. Unless otherwise specified, all uses of the term "subtree" refer to "induced rooted subtree". P-trees $T$ and $S$ are *isomorphic* iff $T$ is the subtree of $S$ and $S$ is the subtree of $T$ simultaneously.

DEFINITION 4 (MAXIMUM COMMON SUBTREE). *Given a graph $G$, $D$ is a P-tree database which pairs with vertices in $G$. $T$ is the common subtree of $D$ such that $\forall t \in D$, $T$ is the subtree of $t$. Furthermore, $T$ is the maximum common subtree of $D$ (denoted as $\Gamma(G)$) if there exists no other common subtree $T'$ such that $T$ is the subtree of $T'$.*

PROBLEM 1 (PCQ). *Given a profiled graph $G(V, E)$, a profile tree $T$, a positive integer $k$, and one query node $q \in G$, find a set $\mathcal{G}$ of graphs, such that $\forall G_q \in \mathcal{G}$, the following properties hold:*

- ***Connectivity cohesiveness**. (1)$G_q \subseteq G$ is connected and contains q, (2)$\forall v \in G_q$, $deg_{G_q}(v) \geq k$;*
- ***Maximal structure**. There exists no other $\tilde{G}_q$ satisfying **connectivity cohesiveness** such that $G_q \subset \tilde{G}_q$ if $\Gamma(G_q)$ and $\Gamma(\tilde{G}_q)$ are isomorphic;*
- ***Semantic cohesiveness**. There exists no other $\tilde{G}_q$ satisfying **connectivity cohesiveness**, such that $\Gamma(G_q)$ is the subtree of $\Gamma(\tilde{G}_q)$.*

# 2. HARDNESS OF THE PROBLEM

## 2.1 Preliminaries

In computational complexity theory and computability theory, a counting problem is a problem only returns the number of all solutions. A counting problem that can be computed by nondeterministic Turing machine running in polynomial time is categorised in the class #P [6] which was firstly introduced by Valiant. Valiant further defined the class #P-complete as the "hardest" problem in #P as the concept of NP-complete is introduced in NP problems. Garey et al. [2] and Papadimitriou et al. [3] proved that if a counting problem is #P-complete, then its associated problem of mining all solutions must be NP-hard. Based on above conclusion, GuiZhen Yang has proved that mining frequent itemsets including subtrees is NP-hard [7]. As for our problem which holds three property defined in Problem 1, all validated communities are required to computed. Thus we follow the same principle that if we can prove that (in worst case) counting the number of all required communities is #P-complete, then our problem is NP-hard.

**Bipartite graph.** A bipartite graph can be denoted as a triple, $G = (U, V, E)$, where vertices can be partitioned to two disjoint sets $U$ and $V$, and $E$ is the set of edges between vertices in $U$ and $V$, i.e., $E \subseteq U \times V$.

A bipartite clique is a subgraph of a bipartite graph such that every vertices in two distinct vertex sets are adjacent. Furthermore, a bipartite clique $G' = (U', V', E')$ is a maximal bipartite clique in a given bipartite graph, if there exists no other bipartite clique $G'' = (U'', V'', E'')$ such that $U' \subseteq U''$, $V' \subseteq V''$, $E' \subseteq E''$ simultaneously.

**Construction of bipartite graph.** Since each node in P-tree is unique and has fixed location in P-tree, knowing all p-tree nodes is enough to reconstruct the original P-tree. Then we can simply construct a bipartite graph $G = (U, V, E)$ from the profiled graph where $U$ is the set of all users in the profiled graph and $V$ is the set of all unique P-tree nodes. Edges in $E$ represent that users in $U$ own the P-tree nodes in $V$. The construction process can be done in linear time. Note that this constructed bipartite graph is not equivalent to the associated profiled graph, because connectiveity of vertices in the profiled graph is not presented in $G = (U, V, E)$. But in the case that the profiled graph is a clique which means each induced subgraph is complete, the profiled graph can be constructed to a bipartite graph without losing generality.

Based on the assumation and this one-to-one correspondence, we can reduce the problem of computing the number of all maximal bipartite cliques containing query node $q$ in the bipartite graph to the problem of computing the number of communities shared maximal common subtrees with $q$ in the profiled graph. The former one is shown in Theorem 1, then the latter one will be proved #P-complete.

## 2.2 Proof

THEOREM 1 ( [4]). *The problem of counting the number of maiximal bipartite cliques in a given bipartite graph is #P-complete.*

Let $C_i(G)$ and $M_i(G)$ denote two sets of qualified communities in which shared maximal common subtrees is unique and $C_c \in C_i(G)$, $|C_c| = i$ and $C_m \in M_i(G)$, $|C_m| \geq i$. Based on Theorem 1, we have Lemma 1.

LEMMA 1. *It is #p-complete to counting the number* $\sum_{i=1}^{|G|} |C_i(G)|$.

Before Now we construct a new bipartite graph $G^+ = (U^+, V^+, E^+)$ followed the strategy in [7] and represent it as a martix. Since the required community contains $q$, the maximal common subtree of the community must be the subtree of $q$. Let $V = \{v_1, v_2, ..., v_n\}$ be the set of $q$'s P-tree nodes and $U = \{u_1, u_2, ..., u_m\}$ be all vertices in the profiled graph. Then, we generate $m$ new items namely $L = \{l_1, l_2, ..., l_m\}$. Let $S(u)$ denotes the subtree of $u$ in $G^+$. $G^+$ is constructed as follows: (1). $U^+ = U \cup S$ where $|S| = m$. (2). for $i \in [1, m]$, $S(u_i) =$

# 3. REFERENCES


[1] V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. *arXiv*, 2003.

[2] M. R. Garey and D. S. Johnson. A guide to the theory of np-completeness. *WH Freemann, New York*, 70, 1979.

[3] C. H. Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[4] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[5] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[6] L. G. Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[7] G. Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 344–353. ACM, 2004.