Appendix for ICDM 2020 paper "Rethinking Local Community Detection: Query Nodes Replacement" Authors are Yuchen Bian, Jun Huan, Dejing Dou, and Xiang Zhang

VI. Proofs

In this section, we provide proofs in the main paper.

A. Proof of Theorem 1

Proof: Suppose we sort nodes based on TP values in the descending order (note that in practice we only need to rank nodes which are local maximum rather than the whole node set). The ranking order for node v_i is denoted as γ_i. If v_i has a local maximum TP value but not the largest, no direct neighbors have larger TP values than $\boldsymbol{\psi}_i$. Then $\forall \gamma_j < \gamma_i, h_{i,j} > 1$. According to Equation (3), $\boldsymbol{\rho}_i > 1$. If there is at least one direct neighbor v_j of v_i has larger TP value, i.e., $\gamma_j < \gamma_i$, since $h_{i,j} = 1$, then $\boldsymbol{\rho}_i = 1$. If $\gamma_i = 1$, then $\boldsymbol{\rho}_i = \max_{\gamma_j > 1} \boldsymbol{\rho}_j \geq 1$ where 1 is got when v_i is the only one local maximum nodes. ■

B. Proof of Theorem 2

Proof:

$$
\begin{aligned}
&|\boldsymbol{\psi}_i - \boldsymbol{\psi}'_i| \\
&= \sum_{l=0}^{D(G)} e^{-(\frac{l}{\sigma})^2} \sum_{v_j \in \Gamma_{i,l}} \mathbf{m}_j - \sum_{l=0}^{\lfloor 3\sigma/\sqrt{2} \rfloor} e^{-(\frac{l}{\sigma})^2} \sum_{v_j \in \Gamma_{i,l}} \mathbf{r}_j \\
&= \sum_{l=0}^{\lfloor \frac{3\sigma}{\sqrt{2}} \rfloor} e^{-(\frac{l}{\sigma})^2} \sum_{v_j \in \Gamma_{i,l}} (\mathbf{m}_j - \mathbf{r}_j) + \sum_{l=\lceil \frac{3\sigma}{\sqrt{2}} \rceil}^{D(G)} e^{-(\frac{l}{\sigma})^2} \sum_{v_j \in \Gamma_{i,l}} \mathbf{m}_j
\end{aligned}
$$

Based on Lemma 5 in [12], we have $\sum_{v_j \in S} (\mathbf{m}_j - \mathbf{r}_j) \leq \epsilon vol(S)$. Then Theorem 2 follows. ■

C. Proof of Theorem 3

Proof: The computation is proportional to the number of neighbors within $\lfloor \frac{3\sigma}{\sqrt{2}} \rfloor$-hops from the set of positive-valued nodes in $\mathbf{r}$ which is denoted as $Supp(\mathbf{r})$. According to Theorem 1 in [12], the number of direct neighbors (i.e., 1-hop) of $Supp(\mathbf{r})$ is less than $\frac{1}{\alpha\epsilon}$. For one hop farther, the number is just multiplied by a factor of d_{avg}. ■

VII. Experiments

In this section, we conduct more experiments to verify the effectiveness of the proposed query-oriented core nodes detection algorithm on local community detection. Except for the case where multiple query nodes are given, we show experimental results for the other two intractable cases where the query nodes are in community boundary and overlapping region. We also provide evidences demonstrating the necessity of each step of the proposed ATP method.

A. A Query Node in Community Boundary Region

From each dataset, we random select 200 distinct query nodes and each of them belongs to exactly one community but has connections with other nodes outside the community. For each query node, we apply ATP to detect associated core nodes. Then for each local community detection method, we compare the detection results between (1) using the original query node and (2) using detected core nodes as new query nodes.

The F-score results for the four datasets are shown in Figure 7. For each dataset (e.g., in Figure 7(a) for Amazon dataset), ten methods are listed on the horizontal axis. For each method, the left (black) bar and right (orange) bar represent the F-scores when using the original query node and the detected core nodes, respectively. We can see that for AM, DB and LJ dataset, the core nodes can improve detection performance for all methods. For YT dataset, 8/10 methods show improvements.

We also compute the relative improvement in F-score for each method. There are averagely 15.01% relative improvement in F-score for query nodes in community boundary region of the four datasets.

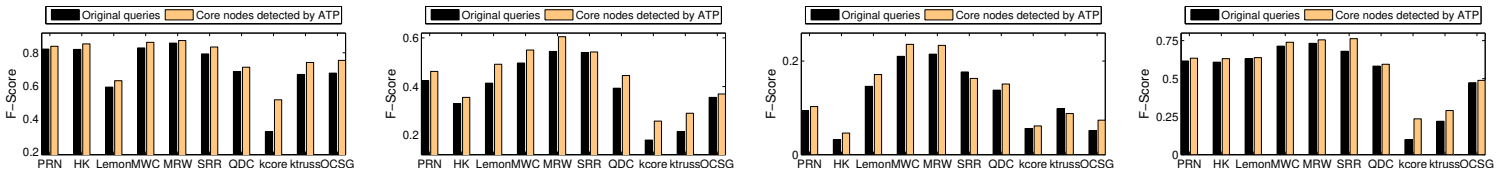
B. A Query Node in Community Overlapping Region

From each dataset, we random select 200 distinct query nodes and each of them belongs to more than one community. For the dataset we use, base on Table I, more than 90% of AM nodes, 30% of DB nodes, 60% of YT nodes, and 60% of LJ locate in community overlapping region.

The F-score results are shown in Figure 8. We can see that more improvements occur when query nodes are from overlapping region than when query nodes are from boundary region. Because most methods can not detect multiple communities for a single query node which is from communities overlapping region. However, the detected core nodes by ATP can separately exist in multiple communities corresponding to the query node. Then local community detection method can find different communities for different set of core nodes. As a result, more accurate communities are obtained instead of just one community using the original query node. There are averagely 25.83% relative improvement.

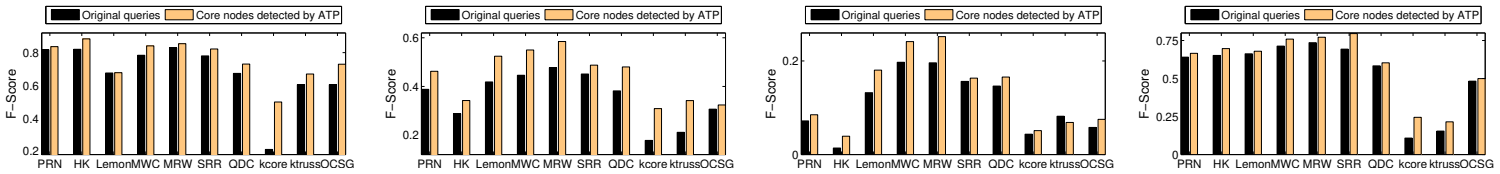
C. Grouping Strategy for Multiple Query Nodes

We also evaluate the accuracy of grouping strategy with Normalized Mutual Information (NMI). NMI checks whether the 100 query nodes are correctly grouped into 25 groups compared with the ground truth. The NMI equals 1 if and only if the groupings are identical with the ground truth, whereas it is 0 if they are independent. Figure 10 shows NMI results of grouping for the four datasets. For AM, DB and LJ, NMIs are above 0.92. For YT, the NMI result 0.81 is due to its star-like community structure. The high NMI results demonstrate that ATP-detected core nodes can truly help group query nodes which are from the same community into one group. This results in both effectiveness and efficiency improvements.



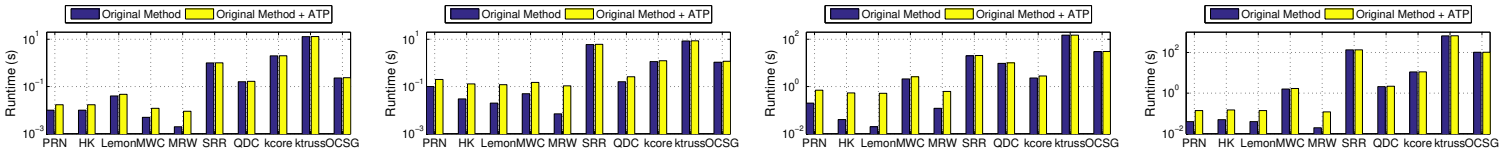
(a) Amazon dataset (b) DBLP dataset (c) YouTube dataset (d) LiveJoural dataset

Fig. 7. Local community detection results for query nodes in community boundary region



(a) Amazon dataset (b) DBLP dataset (c) YouTube dataset (d) LiveJoural dataset

Fig. 8. Local community detection results for query nodes in community overlapping region



(a) Amazon dataset (b) DBLP dataset (c) YouTube dataset (d) LiveJoural dataset

Fig. 9. Runtime per query node (in Section VII-A and VII-B)

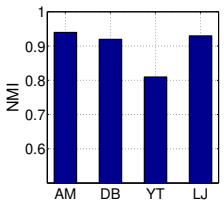


Fig. 10. NMI of grouping multiple query nodes in Section IV-A1

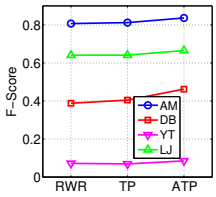


Fig. 11. F-Score results for each step of ATP (using PRN)

D. Evaluation of Each Step of ATP

We also check the necessity of each ATP step in the case of multiple query nodes. In Figure 11, "RWR" means we directly use RWR scores without TP and amplifier computation, "TP" means no amplifier. We can see that each step in ATP is necessary to provide performance improvements.

E. Running Time for Single Query Cases

For single query: In Section VII-A and VII-B, we detect local community for each query node. Figure 9 shows runtimes (per node) of the original methods (blue bars) and modified methods with ATP (yellow bars). In each subfigure, We can see that for the left five methods (PRN, HK, LEMON, MWC and MRW) which are originally fast, ATP brings in extra runtime. But runtimes restrict in the same magnitude. However, ATP can improve their effectiveness (F-score) with large extents. For the right five slow methods (SRR, QDC, k-core, k-truss and OCSG), the cost of the ATP process can be ignored. However, ATP results in impressive improvement on effectiveness (Figures 7 and 8).