Supplemental Material of "Community Focusing: Yet Another Query-Dependent Community Detection"

Theorems and Proofs

Theorem 1. *Given a graph $G(V, E)$ and a set of query nodes Q, the β-attention-core G_o containing Q with the largest β contains no negligible vertices.*

Proof. If a subgraph G' of G contains a negligible vertex u, G' is 0-attention-core as the attention of u is 0. Hence we only need to prove that the minimum attention of the vertices in G_o is above 0, i.e., $ma(G_o, Q) > 0$. Denote G_s as the subgraph containing Q and any two query nodes in G_s are connected by a shortest path between them in $G(V, E)$. Assume there exists a negligible vertex v in G_s. Then v must be on a shortest path between two query nodes. Let q_1 and q_2 be the two query nodes. As v is a negligible vertex, there must exist a neighbor t of v such that $dist_G(q_1, t) = dist_G(q_1, v) - 1$ and $dist_G(q_2, t) = dist_G(q_2, v) - 1$. Then $dist_G(q_1, q_2) \leq dist_G(q_1, t) + dist_G(q_2, t) < dist_G(q_1, v) + dist_G(q_2, v)$. It contradicts that v is on one shortest path between q_1 and q_2. Hence G_s contains no negligible vertices and it is a β'-attention-core with $\beta' = ma(G_s, Q) > 0$. As G_o has the largest $\beta = ma(G_o, Q)$, $ma(G_o, Q) \geq \beta' > 0$. □

Theorem 2. *The attention of a vertex is node-monotonic non-increasing.*

Proof. Given a graph $G(V, E)$ and the query nodes Q, for a vertex v in a subgraph G_s, the only changing factor of $attention_{G_s,Q}(v) = qc_{G,Q}(v) \cdot \sum_{u \in N_{G_s}(v)} w_G(u, v)$ is $N_{G_s}(v)$ (the set of v's neighbors in G_s). In any induced subgraph H of G, as $N_H(v) \subseteq N_G(v)$, $attention_{H,Q}(v) \leq attention_{G,Q}(v)$. □

Theorem 3. *OPT-ma runs in $O(|Q|m + nm' + mlogn)$ where $n = |V(G)|$, $m = |E(G)|$, and m' ($m' \ll m$) is the number of visited edges to check whether the query nodes are connected in a subgraph.*

Proof. By applying $|Q|$ breadth-first search, computing the attention for each vertex (Step 1) can be done in $O(|Q|m)$. In each loop (Step 2), the connectivity of Q should be checked. Usually, the nodes in Q are close to each other. Hence the check can be finished in $O(m')$ ($m' \ll m$). The number of iterations is smaller than n. Deleting u with minimum attention and adjusting the attentions of u's neighbors takes $O(mlogn)$ totally. Therefore, the total time is $O(|Q|m + nm' + mlogn)$. □

Theorem 4. *OPT-cd runs in $O(m_s n_s^2)$ where $n_s = |V(G_s)|$, $m_s = |E(G_s)|$.*

Proof. In each iteration, the removal and maintaining the β-attention-core takes $O(m_s)$ for each vertex. Hence it takes $O(n_s m_s)$ for each iteration. As OPT-cd iterates at most n_s times, it runs in $O(m_s n_s^2)$. □

Theorem 5. *Denote t as the number of iterations (Step 3) in LCF and $G'(V', E')$ as the largest G_{can} during the loops. Then LCF runs in $O(mlogn + t|Q|m' + tm'n')$ where $m = |E(G)|$, $n = |V(G)|$, $m' = |E(G')|$, and $n' = |V(G')|$.*

Proof. Computing the Steiner tree (Step 1) runs in $O(mlogn)$. In each loop (Step 3), expansion (Step 3.1) can be finished in $O(|Q|m')$, and Steps 3.2 and 3.3 runs in $O(|Q|m' + m'n' + m'logn')$. □

Theorem 6. *For each edge $(q_s, q_t) \in E(G')$, $D'((q_s, q_t)) = D_1((q_s, q_t)) = dist_G(q_s, q_t)$.*

Proof. Denote E_{st} as the set of edges that for each edge $(s, t) \in E_{st}$, $s \in n(q_s), t \in n(q_t)$. In our strategy, once an edge $(v_s, v_t) \in E_{st}$ is visited, the edge (q_s, q_t) is created in G' with length equal to $dist_G(q_s, v_s) + 1 + dist_G(q_t, v_t)$. We only need to prove that $dist_G(q_s, v_s) + dist_G(q_t, v_t) = min\{dist_G(q_s, v_x) + dist_G(q_t, v_y) | v_x \in n(q_s), v_y \in n(q_t)\}$.

Assume $\exists (v_m, v_n) \in E_{st}, v_m \neq v_s, v_n \neq v_t$ such that $dist_G(q_s, v_s) + dist_G(q_t, v_t) > dist_G(q_s, v_m) + dist_G(q_t, v_n)$. Without loss of generality, assume v_s is visited before v_t in the breadth-first search.

When v_m is visited before v_n, as the edge (v_s, v_t) is visited before (v_m, v_n), $dist_G(q_s, v_s) \leq dist_G(q_s, v_m)$. As $dist_G(q_s, v_s) + dist_G(q_t, v_t) > dist_G(q_s, v_m) + dist_G(q_t, v_n)$, $dist_G(q_t, v_t) \geq dist_G(q_t, v_n) + 1$.

If q_s is visited before q_t, we have

$$\begin{aligned} dist_G(q_s, v_s) &= dist_G(q_t, v_t) \\ dist_G(q_s, v_m) &= dist_G(q_t, v_n) \end{aligned} \tag{1}$$

Otherwise, Equation 2 is satisfied,

$$\begin{aligned} dist_G(q_s, v_s) &= dist_G(q_t, v_t) - 1 \\ dist_G(q_s, v_m) &= dist_G(q_t, v_n) - 1 \end{aligned} \tag{2}$$

In both cases, $dist_G(q_s, v_s) \geq dist_G(q_s, v_m) + 1$ contradicts with $dist_G(q_s, v_s) \leq dist_G(q_s, v_m)$.

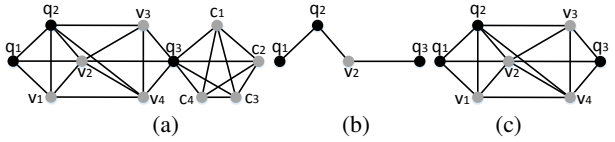


Figure 1: A toy example for explaining LCF. Given the graph $G(V, E)$ and query nodes $Q = \{q_1, q_2, q_3\}$ (black vertices) in (a), (b) is the Steiner tree containing Q, and (c) is the resulted subgraph of LCF.

Similarly, when v_n is visited before v_m, $dist_G(q_t, v_t) \leq dist_G(q_t, v_n)$ contradicts with $dist_G(q_t, v_t) \geq dist_G(q_t, v_n) + 2$. □

Theorem 7. *A minimum spanning tree of G' is a minimum spanning tree of G_1.*

Proof. G' is a connected subgraph of G_1. (1) $V(G') = V(G) = Q$. (2) For each edge $(q_s, q_t) \in E(G')$, $D'((q_s, q_t)) = D_1((q_s, q_t)) = dist_G(q_s, q_t)$ (See Theorem 6). (3) For each edge $(q_m, q_n) \in E(G_1) \setminus E(G')$, due to the breadth-first search in our strategy, $dist_G(q_m, q_n) \geq max\{dist_G(q_x, q_y)|(q_x, q_y) \in E(G')\}$.

Denote $l = max\{dist_G(q_x, q_y)|(q_x, q_y) \in E(G')\}$. T_1 is a minimum spanning tree of G_1. Then $E(T_1) \setminus E(G') \subseteq E(G_1) \setminus E(G')$ and $\forall (q_m, q_n) \in E(T_1) \setminus E(G')$, $dist_G(q_m, q_n) \geq l$. The edges in $E(T_1) \setminus E(G')$ can be replaced with the edges in $E(G')$ as follows:

(1) Find an edge $(q_u, q_v) \in E(T_1) \setminus E(G')$. Remove the edge from $E(T_1)$. Denote T_2 as T_1 with the edge (q_u, q_v) removed. Then T_2 contains two connected subgraphs, and q_u and q_v are disconnected in T_2.

(2) Denote $E(P)$ as the set of edges on one shortest path P from q_u to q_v in G'. There must exist an edge $e \in E(P)$ making T_2 connected with $E(T_2) \cup \{e\}$ (Otherwise, all the vertices lying on P should belong to one connected subgraph of T_2, which contradicts the fact that q_u and q_v are disconnected in T_2). Add e into $E(T_2)$.

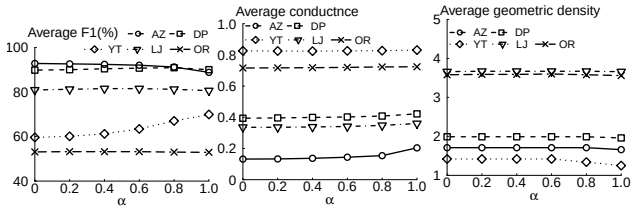
(3) As the length of e is no more than l, and $dist_G(q_u, q_v) \geq l$, the total length of T_2 is no more than T_1. T_2 is also a minimum spanning tree of G_1. Replace T_1 with T_2 and repeat (1).

Using the above replacement, we can obtain a minimum spanning tree T of G_1 such that $E(T) \subseteq E(G')$. As $V(G') = V(G_1) = Q$, and for each edge $(q_s, q_t) \in E(G')$, $D'((q_s, q_t)) = D_1((q_s, q_t))$, a minimum spanning tree of G' is a minimum spanning tree of G_1.

□

Theorem 8. *The speed-up strategy runs in $O(|E'|log|V'| + |Q|^2)$ where E' is the set of visited edges, and V' is the set of visited vertices. Q is the set of query nodes.*

Proof. Step B' takes $O(|Q|^2 + |E'|)$ totally. For each query node q, a set $set(q) = \{q\}$ is created. Once two query nodes $q_1, q_2 \in Q$ are adjacent, the set of q_1 and the set of q_2 are updated as $set(q_1) \cup set(q_2)$. For a query node $q \in Q$, if $|set(q)| \geq |Q|$, G' is connected. Merging the sets of the query nodes runs in $O(|Q|^2)$. $|E'|$ is the number of visited edges in Step B'. Hence Step B' runs in $O(|Q|^2 + |E'|)$.



(a) Average F_1 (%) (b) Average conductance (c) Average geometric density

Figure 2: The performance of FLCF with different α

Benefiting from Steps A' and B', Step C and Step D run in $O(|E'|log|V'| + |E'|)$ respectively. □

Running Examples

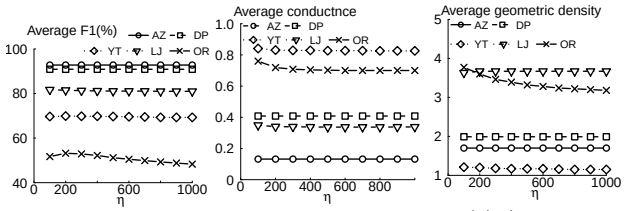
An example of LCF. Given the graph $G(V, E)$ and $Q = \{q_1, q_2, q_3\}$ in Fig.1(a), we assume $\alpha = 0.6$. Firstly, the Steiner tree T ($Q \subseteq V(T)$) shown in Fig.1(b) is built. q_1 is the vertex with the minimum attention in T. $ma(T, Q) = wrd_{T,Q}(q_1) = \frac{1}{1+3} = 0.25$. $cd(T, \alpha) = \frac{2 \cdot 3}{4 \cdot 3^{0.6}} = 0.78$. Thereafter, vertices v_1, v_3, v_4 are added into T as their attentions are larger than 0.25. For example, $ma_{G[V(T) \cup \{v_3\}], Q}(v_3) = \frac{3}{2+1+1} = 0.75 > 0.25$. Denote G_{can} as the subgraph induced by $V(T) \cup \{v_1, v_3, v_4\}$ from G, i.e., $G_{can} = G[V(T) \cup \{v_1, v_3, v_4\}]$. After enlargement (Steps 3.2 and 3.3 of LCF), $ma_{G_{can}, Q}(q_3,) = ma(G_{can}, Q) = \frac{3}{2+2} = 0.75$, $cd(G_{can}, \alpha) = \frac{2 \cdot 15}{7 \cdot 6^{0.6}} = 1.46$. Since there are no vertices around G_{can} with attention no less than $ma(G_{can}, Q)$, $ma(G_{can}, Q)$ and $cd(G_{can}, \alpha)$ can no more be enlarged. G_{can} is returned as the result.

An example of the speed-up strategy. Given the graph $G(V, E)$ in Fig.1(a), assume the set of query nodes $Q = \{q_1, q_2\}$. The auxiliary graph $G'(V', E', D')$ is initialized as $V' = \{q_1, q_2\}$, $E' = D' = \emptyset$. $n(q_1) = \{q_1\}$, $n(q_2) = \{q_2\}$. q_1, q_2 are pushed into $Queue$ with $dist_G(q_1, q_1) = dist_G(q_2, q_2) = 0$. The set of visited vertices $S = \{q_1, q_2\}$. After the initialization, q_1 is popped from $Queue$. v_1, v_2, and q_2 are q_1's neighbors. As v_1 and v_2 are not visited, v_1 and v_2 are pushed into $Queue$ with $dist_G(q_1, v_1) = dist_G(q_1, v_2) = 1$. As q_2 is visited, and $q_2 = s(q_2) \neq s(q_1) = q_1$, q_1 and q_2 are adjoined with $dist_G(q_1, q_2) = 1$ in $G'(V', E', D')$. Thereafter, G' becomes connected. $G'(V', E', D')$ with $V' = \{q_1, q_2\}$, $E' = \{(q_1, q_2)\}$, $D'((q_1, q_2)) = 1$ is returned as the input of Step C. Notably, $T_1 = MST(G') = G'$ and only q_1, q_2, v_1, v_2 are visited in this process.

Parameter Evaluation

The parameters of FLCF are evaluated. α is the tuning factor of the combinational density. η is the size constraint of a resulted subgraph. For each real network, 1,000 communities are selected from the ground-truth communities using the drawn-by-drawn method. For each community, $|Q|$ nodes are selected as a query. Then the quality of the results is evaluated using F_1, conductance, and geometric density.

Fig.2 shows the performance of FLCF with α ranging from 0 to 1. η is set to 200 in this experiment. Fig.2(a) reports the average F_1. F_1 varies with α, which shows that the density of ground-truth communities in each network is



(a) Average F_1 (%) (b) Average conductance (c) Average geometric density

Figure 3: The performance of FLCF with different η

different. F_1 of Youtube increases with increasing α. This is because the ground-truth communities of Youtube are sparse. When α becomes larger, FLCF tends to find a small subgraph containing the query nodes, which increases F_1. Fig.2(b) shows the average conductance. As α increases, the combinational density tends to be the internal density, which is sensitive to the number of vertices. As a consequence, some vertices densely connected to the query nodes are removed. Therefore, the conductance becomes larger. Fig.2(c) presents the average geometric density. In most networks, the geometric density are stable, which confirms that FLCF can find densely connected subgraphs. As the ground-truth communities in Youtube are sparse, FLCF with large α ($\alpha \geq 0.8$) tends to remove more vertices in a dense subgraph, which decreases the geometric density.

Fig.3 presents the performance of FLCF with η ranging from 100 to 1,000. For each network, α is set to the number which achieves the highest F_1 in Fig.2(a), i.e., α is set to 0.0, 0.8, 1.0, 0.4, and 0.2 for Amazon, DBLP, Youtube, LiveJournal and Orkut respectively. Except for Orkut, the performance of FLCF is stable with different η, which shows that 200 is large enough to get a community meaningful to the query nodes. In Orkut, communities overlap each other heavily. As η increases, more vertices are discovered by LCF, which leads to the decrease of F_1 on Orkut. As η increases, the resulted subgraph becomes larger and fewer edges are pointing out of the subgraph. Hence the conductance of Orkut decreases. Due to the β-attention-core which removes the vertices loosely connected to the query nodes, conductance becomes stable when $\eta \geq 200$. As η increases, the size of the resulted subgraph becomes larger. As the geometric density is more sensitive to the number of vertices than the number of edges, the geometric density decreases in Orkut.

As the most time-consuming step of FLCF is building a Steiner tree, the time costs of FLCF with different α and η are similar to the time costs of LCF on the real networks.