A Review on: Incremental Closeness Centrality in Distributed System

Md Maruf Hossain

I. Closeness Centrality

The closeness centrality of vertex defines how close a vertex to all other vertices. The closeness centrality of a vertex u can be measured using the total number of vertices divided by the sum of the shortest path of all other vertices of the graph G from the particular vertex u. One can use BFS to calculate the shortest path that will cost $O(n+m)$. So, for the total n vertices, the overall complexity is $O(n(n+m))$ for the offline version of the closeness centrality. But in a dynamic graph an edge can be inserted or deleted any time from the graph and to measure the closeness centrality of this kind of graph is called incremental closeness centrality.

II. Contributions

The authors of these papers [1]–[3] introduced a Nobel technique that can calculate the incremental closeness centrality only for the affected vertices. Their proposed filter can successfully filter out vertices that require to update the centrality score.

- Lavel-based: if an edge need to insert between two vertices u and v then the only vertices need to updated their cenjqwwjtrality score can be filtered out by $|d_G(s,u) - d_G(s,v)| > 1,\ s \in V - u, v$
- They maintain identical vertex lists that a vertex can represent a group of vertices that are similar to the representative vertex.
- Biconnected component decomposition(BCD): They calculated the biconnected component after insertion or deletion of an edge in the graph and proved that it is sufficient to perform SSSPs only on one of the biconnected component that contains the edge. Other vertices need to updated their centrality score if their representative is an articulation vertex of the biconnected component that is holding the new edge.
- *SpMM* formulation: In their work, they try to represent the BFS problem as a *SpMM* formulation. They also mentioned that BFS only handles the vertices for a particular level, so it is not sufficient for a tradition queue based BFS.

References

[1] Ahmet Erdem Sariyüce, Erik Saule, Kamer Kaya, and Ümit V Çatalyürek. Streamer: a distributed framework for incremental closeness centrality computation. In *2013 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–8. IEEE, 2013.

[2] Ahmet Erdem Sariyüce, Kamer Kaya, Erik Saule, and Ümit V Çatalyiirek. Incremental algorithms for closeness centrality. In *2013 IEEE International Conference on Big Data*, pages 487–492. IEEE, 2013.

[3] Ahmet Erdem Sarıyüce, Erik Saule, Kamer Kaya, and Ümit V Çatalyürek. Incremental closeness centrality in distributed memory. *Parallel Computing*, 47:3–18, 2015.