Keyword-Aware Top-r Influential Communities

Shamik Roy
Md. Saiful Islam

Contents

1 PROBLEM STATEMENT

Community search in a large social networks has been an active research topic in the the last decade. Li introduced the notion of influential community and find top-r k-influential communities based on the concept of k-core that captures the influence of a community. In this case, a community is defined as a connected subgraph in which each node has degree at least k, and they find top-r such communities where the influence of a community is defined as the minimum weight of the nodes in the community. They have modeled the weight of a node as a numerical value that signifies the influence.

We introduce the concept of keyword-based communities where each user/node can be described as a set of keywords and associated weights denoting her expertise and her influence in a particular topic, respectively. So for a given set of query keywords, our focus is to identify top-r communities. We also re-define the concept of influential communities by introducing the edge-weights on the graph that cover a wider range of problem. For example, if two authors write many papers together they are more likely to be the member of the same community than that of the two authors who wrote only one paper together. To incorporate the edge-weights, we need to model the community as a (k,w)-core subgraph, where each member of the community is connected with at least k other members of the group and has a minimum interaction weight w with these members.

Given an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Every node is associated with some (keyword,influence score) tuples and every edge is associated with an weight. Let, $W(v_1, v_2)$ denotes the weight of the edge connecting vertex v_1 and vertex v_2. Let, $d(v_1)$ denotes the influence value of vertex v_1. We need to find the top-r influential communities.

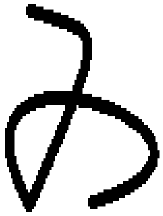
A natural variation of this problem is to consider a dynamic graph (or stream) where edge weights, and node weights and topics change. For example, when two authors write a new paper the edge weight between them changes, and also based on the topic of the paper, the weight of nodes on different topic may change. In such a scenario, finding top-r communities and constantly updating the top-r communities for a dynamic graph can be an interesting topic in this research.

2 DIFFERENCES WITH PREVIOUS WORKS

2.1 NOT TAKING INPUT OF 'k' FROM USER

We are not taking any input of k-core from user(as Li). Because cases may arrive that user searching for a specific k-core may not present in the entire graph and practically, users may not have the idea of how correlated (idea of k) the existing communities are. So, we eliminated the idea of finding top-r k-influential community rather made the k-core thing dynamic to deliver the user top-r influential community.

Figure 1: Sample graph.



EXAMPLE 1. Consider the forest in Figure 1. Here If an user search for a top-5 5-influential community, the result will be null. Because, there is no 5-core community in the forest.

Moreover, in real life, users are mostly interested in top influential communities rather in the connectivity among the members of the community.

2.2 MIXTURE OF VARIOUS CORES IN RESULT

As we are not taking any input for 'k' from user, the algorithm cannot bind itself in finding a specific core. Because of considering edge weight as an influence factor,the final ordering of influential communities becomes fully independent of core order.

EXAMPLE 2. Consider the forest in FIGURE 1. In this graph a 2-core influential community consists of the nodes $\{1, 2, 3, 4\}$. Another 3-core influential community consists of the nodes $\{9, 10, 11, 12\}$. Despite of having more connectivity the 3-core influential community may lag behind the 2-core community. Because of considering edge weights the 2-core influential community may yield greater influential value than the 3-core influential community.

3 Basic Solution Approach

Firstly, we want to give a solution with some relaxation of the problem ignoring the edge weight. The relaxed problem statement is:

Given an undirected graph $G(V, E)$ where, each node vϵV represents an individual person containing a set of (keyword, influence score) tuples indicating the influence of the person in the field denoted by the keyword. And edge eϵE exists between two nodes (u,v) if person u and v are socially acquainted with each other. For example, u and v together may write a research paper.
Now given a query q = (r,x_1,x_2,...,x_n,R) where x_1,x_2,...,x_n are the query keywords and R is a relational operator (either AND query or OR query). We need to find the top r - influential communities.