Finding Densest Subgraph

Prateek Chanda

May 2020

1 Concept of dense subgraph

While there are many proposed density metrics discussed in previous research works, including edge density.

One of the popular metrics for calculating density of a graph is given by

$$d_G = \frac{2E}{(V(V-1))}$$

However, this leads to many trivial cases, for example two nodes with a single edge connecting them has high density.

Also a wider sense of how dense a subgraph can be given by the measure of how close the given subgraph is to a clique consisting of the same nodes.

The total number of edges in the subgraph, given its adjacency matrix as A is given by $x`Ax$ where x is the indicator vector of the nodes in the subgraph.

Hence the measure how close a subgraph is close to its equivalent clique is given by Density Distance DD_G

$$DD_G = ||x`(1_{mxm} - I_{mxm})x - x`Ax||$$

Hence, we currently propose the following density for a particular subgraph G

$$d_G = \frac{1}{DD_G}$$

We modify d_G to the following

$$d_G = \frac{|E|}{DD_G}$$

This is done in order for our definition of density have a balance of good no. of edges as well being as close to its corresponding clique.

Since otherwise, K_2 will be the densest subgraph and the algorithm will keep iterating till it reaches K_2 which is trivial.

Let V is the universal set of nodes and E be the universal set of edges in the original graph, $G_{V,E}$.

Let S be any subgraph such that $S \in G_{V,E}$.
Objective :

$$max\ d_S$$

$$\forall S \in G_{V,E}$$

2 Proposed methodology

```
1. Start with initial graph $G_{V,E}$ and calculate density
2. Check which node is the contributing factor for not the graph being
  closer to its clique's version. This can be computed by check for which
  entry in the matrix, the difference between its clique and current
  Adjacency matrix is maximum.
3. Remove that node. If after removing density is higher than before
  continue else stop algorithm
4. Continue step (1-3) until the entire graph gets disconnected into two
  connected subgraphs, say $G_t(V_t, E_t)$ and $G_s(V_s, E_s)$
5. Compare w.r.t Comparative Density Score, whether $G_t$ has a
  higher score or vice-versa.
if $CDS(G_t) > CDS(G_s)$ then
  | Repeat steps (1-4) for $G_t$
else
  | Else Repeat steps (1-4) for $G_s$
end
```

2.0.1 Comparative Density Score

Given two subgraphs, created from disconnecting graph G, say $G_t(V_t, E_t)$ and $G_s(V_s, E_s)$, the comparative Density Score provides an estimate which of the subgraph contains a denser subgraph within it.

The concept helps to only search inside a particular subgraph instead of searching in both the subgraphs, a classic example of divide and conquer example (solving only a particular subproblem as in Binary Search).

Hence this drastically reduces the total time complexity of the algorithm, since at each stage of the division, we need to work on only one subproblem.

The **Comparative Density Score** for a subgraph $G_t(V_t, E_t)$ is given by,

$$CDS(G_t) = \lambda \frac{(V_t)(V_t - 1)}{2} + (1 - \lambda) DD_{G_t}$$

where is a balancing parameter.

The second term indicates the corresponding measure of how close the subgraph is to its corresponding clique.

While the first term indicates the maximum probable edges that could be possible in the subgraph $G_t(V_t, E_t)$.

2.0.2 Theoretical Guarantees of CDS

Lemma 1: S belonging to G will always have $DD_S < DD_G$ since DD_S is contained in DD_G

Lemma 2 : $DD_{S_t} > DD_{S_s}$ indicates S_t has a higher chance of a dense subgraph within it, given S_t and S_s has equal distribution of nodes.

Lemma 2 would not hold everytime when two subgraphs do not have equal no. of nodes.

Hence we introduce the concept of λ. Considering the example of one subgraph K_2 and another a large subgraph containing say 7-8 nodes, only DD_G will chose K_2. Hence using lambda we give importance to the corresponding two items, to make a wiser decision.

Given two subgraphs $G_t(V_t, E_t)$ and $G_s(V_s, E_s)$ if $V_t > V_s$, $CDS(G_t)$ is calculated with a proportionate smaller value of λ and hence more importance given to DD_G.

Similarly, for $CDS(G_s)$, its calculated with a proportionate large value of to take into account higher importance of the first term, rather than the second term.

Basically give higher importance to the term for a graph, where its comparatively lagging than the other and lower importance to the term that is having advantage.

2.1 Multiple edges

For consideration of multiple edges (for our use case), we define the **Distance Density** measure by the following

$$DD_G = ||x`(C * 1_{mxm} - I_{mxm})x - x`Ax||$$

where C is the maximum occurrence of an edge among all the edges in the edgeset for a particular graph.