Gmail **Saby S <ssahoo.infinity@gmail.com>**

Queries related to your NIPS 2017 paper "Thy Friend is My Friend"

Christina Lee <celee@mit.edu> Tue, Jan 23, 2018 at 2:07 AM
To: Saby S <ssahoo.infinity@gmail.com>
Cc: Jennifer Chayes <jchayes@microsoft.com>, Christian Borgs <Christian.Borgs@microsoft.com>, Devavrat Shah <devavrat@mit.edu>

Hello Sabyasachi,

Below are the answers to your questions.

- I understand that *r* is a hyper-parameter chosen by us. I wanted to clarify how exactly are you defining *r* or the value s (shortest distance between user (or vertex) u and item i \in S_ {u, s} in step 2 of your algorithm on Page 6? For example, for another movie (say *M2*) rated by the direct friend (*U2*) of a user (*U1*) (i.e. friend who has rated a movie (*M1*), also rated by the user), do we say that *M2* is at *r* = 1 hop (considering *M1* is directly rated by *U1 is* at *r* = 0 hop) or is *M2* at *r* = 3 hop (since there are three intermediate ratings/edges between the user and the movie, i.e. *U1-M1, M1-U2, U2-M2*)? This may not matter in an exact sense, but I am asking just for clarification.
  In this case, M2 would be a r=3 becase there are 3 ratings/edges between the user and movie. It is essentially the same as the shortest path in the bipartite graph where edges are ratings.

- Also, we might get the product-ratings for some movies at a r - hop distance, which were also directly rated by the user (despite having no loops in the path from the user to the movie). In such cases, should we consider the ratings that we get at a r - hop distance from the user or the ratings directly given by the user or simply consider no ratings at all for the movies directly rated by the user?
  Since we grow the neighborhoods around each vertex according to a breadth first tree, every movie (and user) appears only once in the tree. If a movie is directly rated by the user, then it will be at radius/distance 1, and thus does not appear again at a farther distance r-1. However, this rating is still used in the computation because the components of the vector N_{ur} consists of the product of ratings along the path from the root to distance r. Therefore, there will be many "descendants" of this directly rated movie, such that the product of ratings along the path will include the rating provided directly by the user for the movie at distance 1. To gain some intuition, consider comparing a pair of users U1 and U2 which have directly commonly rated a movie M1. Then the descendants of M1 in the BFS trees rooted at U1 and U2 will have a high overlap, such that the entries in (N_{u1,r} - N_{u2,r}) that correspond to descendants of M1 will have a value of the difference between ratings that U1 and U2 provided to M1 multiplied by the path from M1 to the corresponding descendant.

- In step 2 of the algorithm, your algorithm creates a BFS tree rooted at every vertex. Does building DFS tree alters anything about the algorithm?
  Our analysis does rely on the BFS tree to make sure that the growth of the neighborhood is "even". For example, suppose that the graph were fully connected, then growing a DFS tree would mean that the root node has only a single child since all vertices would be already explored by the DFS through a single initial branch. In our analysis we use the "even growth" to guarantee that each depth/layer of the tree has sufficiently many nodes for us to provide concentration results for the growth of the neighborhood at each layer. The DFS would mean that the earlier visited nodes would have more children and descendants than later visited nodes, thus also introducing bias.

Sincerely,
Christina Lee
[Quoted text hidden]