WMR 2015 Final Term

May 22, 2015

Assignment

The Final Term consists in an advanced study of the Last.fm social network sample already collected and discussed for the mid term assignment. In detail, each group has to choose three exercises (among the proposed ones), solve and discuss them producing a final report.

Please, send the reports to pedre@di.unipi.it, giulio.rossetti@isti.cnr.it, lpappalardo@di.unipi.it in pdf format (use as subject [WMR2015 Final Term]). The Final Term projects will be discussed during an oral dissertation.

Oral Exam: Agree a date for the oral dissertation with the professor.
Submission: the project must be submitted within the summer exam session (at least 3 days before the date of the discussion).

Exercises

1. **Community Discovery**
   Apply to the crawled network at least two community detection algorithms among: K-Cliques, Girvan-Newman [1] and DEMON [2]. Discuss the obtained results (distribution of nodes in communities, number of communities, average density of communities, etc. etc.) and validate the communities using external data such listenings, artists, genres, hotness. Is there a prevalent genre/artist/song listened by users in the same community?

2. **Node Similarity (tie strength)**
   Define some measures of similarity among nodes using information related to their listenings. Consider such "similarity" measures as proxies for approximating tie strength and study:

   - their correlation with other pairwise network measures (i.e., common neighbor, adamic adar, jaccard, edge betweenness);
   - how the network structure is affected by the removal of edges for both ascending and descending values of tie strength.

3. **Epidemics**
   Implement and test the SIR, SIS or SIRS models explained in [3] (chapter 21: Epidemics). Alternatively apply a cascade model (as introduced in [3] chapter 19: Cascading behaviors in networks).

4. **Local Diffusion**
   Following the approach proposed in [4], select a subset artists (up to five) having different degree of hottness and for each one of them:

   - identify those users which are the first in their neighborhood to listen his/her songs (we will refer to them as *Leaders*);
   - for each (Artist, Leader) build the tree rooted in the Leader and connecting those nodes which have listened Artist imposing a temporal constraint on the edges;
   - for each (Artist, Leader) compute the depth of the identified tree and the width at its first level (i.e. the number of nodes that listen Artist and are at one-hop from Leader);
   - discuss if there exist Artist-characteristic width and depth for the local diffusion trees.
   - compute the distribution of tree size (number of nodes in the tree) generated by the Leaders identified. Make also the distribution of the height of the trees generated by the Leaders.

5. **Induced Graph analysis**
   Build the Artist-induced network: in such graph two artists are connected iff they are listened by users that share a link on the original graph. Each

edge connecting a pair of Artist is thus, weighted with the number of links in the social graph which concur to its creation. Reproduce on such network the analysis performed to address the other problem selected.

6. **Diminishing returns**
   How many friends have to listen to a music genre before I start to listen to that music genre? Given a music genre g, try to answer this question by computing, for each user A in the social network, the following value $t_A(g)$: the number of friends who listened to music genre g before A. Make it for all the main music genres (pop, jazz, punk, folk, rock etc.). Finally make the distribution of this value across the population of users separately for each music genre. What is the shape of the distribution? Are there interesting difference?
   Example: if Jon starts to listen to rock songs after that two friends of him listened rock songs, then $t_{Jon}(rock) = 2$. If he listens to jazz songs after that four friends of him listened jazz songs, than $t_{Jon}(jazz) = 4$.

Code and libraries

To approach the exercise you will need to write some code: you can use any programming language and library you prefer. However, especially for those who are not familiar with code developing, we suggest to use the networkx Python library due to its simplicity.

Implementations of the DEMON algorithm can be downloaded from:
`http://www.giuliorossetti.net/about/ongoing-works/material/`.
An implementation of k-cliques is offered by networkx[1].

References

[1] Girvan M. and Newman M. E. J.: Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 78217826 (2002)

[2] Michele Coscia, Giulio Rossetti, Fosca Giannotti, Dino Pedreschi: DEMON: a local-first discovery method for overlapping communities. KDD 2012:615-623

[3] David A. Easley, Jon M. Kleinberg: Networks, Crowds, and Markets - Reasoning About a Highly Connected World. Cambridge University Press 2010

[4] Diego Pennacchioli, Giulio Rossetti, Luca Pappalardo, Dino Pedreschi, Fosca Giannotti: The Three Dimensions of Social Prominence, SocInfo 2013

[1] `https://networkx.github.io/documentation/latest/reference/algorithms.community.html`