Centre for Continuing Education
indian Institute of Science

Data Structures and Graph Analytics - Assignment #2
Link Prediction in an Undirected Graph

Due date : 5-Dec-2021, 11.59 PM

This assignment deals with **link/edge prediction** in networks. *You are required to use the SNAP package to implement this asignment in Python.* You should not implement any of the functionalities possible with SNAP using your own code; you should use SNAP. Of course, functionalities unavailable in SNAP need to be implemented in Python. A network is represented as an undirected graph. Such a graph is viewed as $G(V, E)$, where V is the set of nodes and E is the set of edges; *there are no self edges or multiple edges.* Let **degree of node** $x \in V$, $d(x)$ be given by

$$d(x) = |\{e : e \in E \text{ } and \text{ } x \text{ } is \text{ } an \text{ } end \text{ } vertex \text{ } of \text{ } e\}|.$$

Let U be the universal set containing all $\frac{|V|.(|V|-1)}{2}$ possible edges, where $|V|$ is the number of vertices. So, $U-E$ is the set of **nonexistent edges** in G. We would like to predict K **important nonexistent edges** based on **ranking** the elements of $U - E$. Your **implementation has to deal with two parts** as explained below:

- **Part1**:
  - The compressed directory is *contact-high-school-proj-graph.tar.gz* and is available for download from `https://www.cs.cornell.edu/~arb/data/contact-high-school/index.html.` (Note: Do not copy and paste the above URL; it may not work because of the ˜(tilde) character). The file is also attached along with the assignment description.
  - There are two files in the folder/directory. You need to consider the file *contact-high-school-proj-graph.txt* only. There are 5,818 lines in the file; each line corresponds to a weighted edge of the graph in the form i j w where i and j are the two nodes of the undirected edge and w is its weight.
  - This dataset corresponds to a social network of 327 high school students and two nodes have an undirected edge between them if the associated students are friends. Observe that j values are listed in non-decreasing order.
  - Note that there are 327 nodes in the graph. So, $|V| = 327$ and $|E| = 5818$ in G. Convert it into a binary graph by viewing all these w values to be equal to 1. Use this information to store the given undirected graph G using the SNAP package.

- **Part2**: *Use the functions in SNAP to rank each of the nonexistent edges*, that is elements of $U - E$, using the following **three scoring functions and print the K top-raked edges for a given K, in $U - E$, along with their respective scores** in each case. Any edge $e \in U - E$ may be viewed as an ordered pair $< v^1, v^2 >$, where $v^1, v^2 \in V$ are the end vertices of edge e.

  1. **Common Neighbors (CN)**: For edge $e_i \in U - E$, $i = 1, \cdots, |U - E|$, the set of common neighbors of e_i is

$$CN(e_i) = N(v_i^1) \cap N(v_i^2)$$

where $N(v)$ is the set of neighbors of $v \in V$. The **Common Neighbor Score (CNS)** of e_i is given by

$$CNS(e_i) = |CN(e_i)|.$$

For edges $e_i, e_j \in U - E$, e_i is **more important than** e_j if $CNS(e_i) > CNS(e_j)$. *Sort the edges in $U - E$ by non-increasing value of their CNS.*

2. **Adamic-Adar Score (AAS)**: For edge $e_i \in U - E,\ i = 1, \cdots, |U - E|$,

$$AAS(e_i) = \sum_{z \in CN(e_i)} \frac{1}{log\ d(z)},$$

where $d(z)$ is the degree of node z. In this case, e_i **is better than** e_j if $AAS(e_i) \geq AAS(e_j)$. *Sort the edges in $U - E$ by non-increasing value of their AAS.*

3. **Katz's Score (KS):** For edge $e_i \in U - E,\ i = 1, \cdots, |U - E|$, where $e_i =< v_i^1, v_i^2 >$,

$$KS_\beta(e_i) = \sum_{l=2}^{6} \beta^l.|paths^l_{v_i^1, v_i^2}|.$$

where $paths^l_{v_i^1, v_i^2}$ is the set of paths of length exactly l between v_i^1 and v_i^2. *Use a value of 0.1 for* β in computing the score. *Sort the edges in $U - E$ by non-increasing value of their KS_β.*

- **Reference:** *David Liben-Nowell, Jon M. Kleinberg: The link prediction problem for social networks. CIKM 2003: 556-559* (please see the attached file for this assignment).