

**School of Computing Science & Engineering**

**CSE3021 – Social and Information Networks**

**First Review**

**Title of the Project:**

**Link Prediction in Social Networks**

**Student Information:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | Register No. | Name of the Student | Mobile No. | Email ID |
| 1. | 15BCB0049 | DEBASHIS KARMAKAR | 9585589093 | Debashis.karmakar2015@vit.ac.in |

**Abstract**

|  |
| --- |
| **Objective: To understand the Link Prediction problem and propose an improvement to existing solution/algorithm.**  **Problem Statement:**  **Link Prediction Problem**  Different kind of links or edges between the nodes exist in a social network. For example, social contacts, phone-calls or hyper-references. On analysis of social networks, there can be many information about the linkage between the nodes that are not discovered or unknown at a given point of time. Link Prediction is  the problem of predicting links that either dont yet exist at the given time t or exist, but unknown up to this time. Given a picture of a social network(nodes and links) at time t, we need to predict accurately the links that will be added to  the network during the interval from time t to a given future time t+1. In effect, the link prediction problem concentrates on to what extent can the evolution of a social network be modelled by using intrinsic features of the network itself?    **Literature Review: (references given below)**  **These are the present methods available for link prediction:**   1. NODE NEIGHBORHOOD ALGORITHMS    1. The Common Neighbours method : It provide a measure for similarity by calculating the intersection of the sets of neighbours of the nodes to predict future linkage.    2. Jaccards coecient: measures number of the features(neighbors) that are shared between two nodes commensurate to all features that either one of the nodes has.    3. Adamic/AdarIt is a measurement that compares how many attributes two nodes have in common. They rate items that are unique to a few users more heavily than items shared amongst a huge group of users.    4. Preferential Attachment is based on the hypothesis that a node x will get new neighbors faster than a node y given y has less neighbors than x. So the probability that a node will form a new link varies with number of its present neighbors. The likelihood of two nodes being connected by an edge based on preferentialattachment is measured by multiplying the number of their neighbors. 2. PATH BASED ALGORITHMS    1. Katz : A measurement that takes all paths between two nodes in consideration whilerating short paths more heavily.    2. SimRank: If two nodes are referenced by more similar objects, then the two nodes have large similarity value.    3. Hitting Time and Commute Time    4. Rooted PageRank : Rooted PageRang is a modication of the Page Rank measure (which is an attribute of a single vertex) for link prediction.    5. PropFlow and High-Performance Link Prediction    6. Supervised Random Walks 3. META APPROACHES - Meta-Approaches alter the data before being passed to one of the path based approaches.    1. Low-rank approximation    2. Clustering    3. Bayesian Probabilistic Model    4. Linear Algebraic Method   We tried out one of these approaches(Preferential Attachment) and try to modify the same, or come up with a unique approach.  **PROPOSED PROBLEM STATEMENT:**  LINK PREDICTION in SOCIAL NETWORK:  Algorithm chosen: PREFERRENTIAL ATTACHMENT  Innovation: implement REVERSE PREFERRENTIAL ATTACHMENT  **PROPOSED SOLUTION:**  **Method we chose to work on:**  Preferential attachment algorithm    It works on the “Rich gets Richer” princliple  **Modification we tried to make:**    THE MODEL:  Consider a network that evolves by the addition and removal of vertices. In each unit of time, we add a single  vertex to the network and remove r vertices. When a vertex is removed so too are all the edges incident on that  vertex, which means that the degrees of the vertices at the other ends of those edges will decrease. Non-integer  values of r are permitted and are interpreted in the usual stochastic fashion. (For example, values r < 1 can be interpreted as the probability per unit time that a vertex is removed.) The value r = 1 corresponds to a network of  fixed size in which there is vertex turnover but no growth; values r < 1 correspond to growing networks. In principle  one could also look at values r > 1, which correspond to shrinking networks, and the methods derived here are  applicable to that case. However, we are not aware of any real-world examples of shrinking networks in which  the asymptotic degree distribution is of interest, so we will not pursue the shrinking case here. We make two further assumptions, which have also been made by most previous authors in studying these types of systems:  (1) that all vertices added have the same initial degree, which we denote c;  (2) that the vertices removed are selected uniformly at random from the set of all extant vertices. Note however that we will not assume that the network is uncorrelated (i.e., that it is a random multigraph conditioned on its degree distribution as in the so-called configuration model). In general the networks we consider will have correlations among the degrees of their vertices but our solutions will nonetheless be exact.  Let pk be the fraction of vertices in the network at a given time that have degree k. By definition, pk has the normalization.    Our primary goal in this project will to evaluate exactly the degree distribution pk for various cases of interest.  **BASIC algorithm ABOUT OUR SOLUTION(preferential)**  **for i in range(iNodes):**  **dNetwork[i] = []**  **for node in dNetwork.values():**  **fThresh = 1.0 / (iLinks + i + 1) \* (len(node) + 1)**  **if(random.random() <= fThresh):**  **node.append(i)**  **iLinks += 1**  **our logic, in reversing(reverse preferential)**  **j=0**  **k=3#using 3 passes!!!**  **print "Beginning Communist model(reverse-preferential attachment) , we are using ",k," passes"**  **while(j<k):**  **lDegrees1 = [len(node) for node in dNetwork.values()]**  **avg = sum(lDegrees1)/float(iNodes)**  **print "Average Degree after pass ",j," : ",avg**  **j+=1**  **for i in range(iNodes):**  **#avg = sum(lDegrees)/float(iNodes)**  **#print "For node ",i," avg is ",avg**  **for node in dNetwork.values():**  **fThresh = 1.0 / (iLinks + i + 1) \* (len(node) + 1)**  **rd=random.random()**  **if(rd <= fThresh):**  **if(node):**  **node.pop()**  **iLinks -=1**  **des +=1**  **if(rd <= (1-fThresh) and len(node)<avg):**  **node.append(i);**  **iLinks += 1**  **add += 1**  Test cases:   * There are 1000 nodes taken. * Probabilities[0,1] are generated randomly using random.random() * For our tweaked preferential attachment model, we have ran it across 1 pass and 3 passes   RESULTs:   * The change in degree of each of the 1000 nodes is presented as a list(a python data structure) * Average for each pass on each model is calculated , along with the no.of nodes made and destroyed * Finally graphs are plotted x: NODE ID , y: NODE DEGREE * For each model, blue-dot = capitalist(preferential attach) , red-triangle = communist(reverse-preferentail attchment) |
|  |
|  |
| Test case #1:  WE START WITH k=3 // k is the no. of passes  OUTPUT AS OBSERVED:    Average degree : 1.011        Average degree : 3.153    Capitalist model, very few are rich    Communist model, after 3 passes, “poor gets richer”    Combined model, the above 2 graphs are plotted together  Test case #2:  Only 1 pass done        Capitalist model, very few are rich    Communist model, after 1 pass, “poor gets richer”    Combined.  **Evaluation Metrics:**   1. Average Degree distribution 2. Identifying change in degree in specific nodes after passing through the model 3. Check if multiple passes through same model gives incorrect values   **Performance Analysis for the identified metrics:**   * *Average Degree distribution*: The average always increases per pass through our model.   It is observed in the triple pass test case that:  At k=1 first pass , average is 1.011  At k=2 second pass , average is 2.305  At k=3 third pass, average is 3.153   * Change in specific nodes(NODE ID’s) are mentioned in circled colour marks given below * After multiple passes, k=3, node degree observed by our modified model is still less than iNodes(=1000) , *node degree < iNodes-1*   *(*Unlike the faulty model which ma’am had asked to be rectified*)*  **FINAL Result(OUR OBSERVATION):**  We see that in the reverse-preferential attachment model, which we like to call the communist model,  Nodes with degree less than avg degree have a chance to make more links with other node, and nodes more than avg can only cut down on the number of nodes they are connected to.    For a k=1 , single pass model ^  Blue pen marks , node 285 which has fallen from its grace  Red pen marks , node 281 which has managed to make 2 new links!!  Black pen marks, node 270,271 who have not changed their values at all  Similarly, for k=3 triple pass on our communist model, the avg has reached a value close to 4 here.    **Conclusion:**  The communist model described above accurately simulates the fact that people(nodes) when placed in an environment which allows a reverse preferential attachment and gives chance for people(nodes) below average to make links with others, tend to improve the overall average node distribution in the network allowing better resource sharing.  (references given below) |

**References: (Min. 10 references, 1 or 2 Base papers and 9 or 10 supporting Papers/Link)**

|  |  |
| --- | --- |
| **1.** | Exact solutions for models of evolving networks with addition and deletion of nodes  Cristopher Moore,1, 2, 3 Gourab Ghoshal,4, 5 and M. E. J. Newman3, 4 |
| **2.** | <http://www.leonidzhukov.net/hse/2016/networks/papers/NowellKleinberg07linkprediction.pdf> |
| **3.** | https://pdfs.semanticscholar.org/e7d3/0fefe1b99c21813873f976e46d03dc82b4fc.pdf |
| **4.** | http://www1.se.cuhk.edu.hk/~seem5010/slides/kdd2011\_placefriends.pdf |
| **5.** | https://pdfs.semanticscholar.org/9ea5/6f4441ec22e85c8a6077d7e17b7ee88e31cc.pdf |
| **6.** | http://ethesis.nitrkl.ac.in/5217/1/109CS0184.pdf |
| **7.** | http://www.ise.bgu.ac.il/faculty/liorr/conf5.pdf |
| **8.** | https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a020.pdf |
| **9.** | <https://www.researchgate.net/profile/Panagiotis_Symeonidis/publication/221141172_Transitive_node_similarity_for_link_prediction_in_social_networks_with_positive_and_negative_links/links/0fcfd50df15c13977c000000.pdf> |
| **10.** | <http://domino.mpi-inf.mpg.de/intranet/ag5/ag5publ.nsf/448742fce85ecfa580255f0d006b34c7/818fecf002c66936c12575fd00685881/$FILE/main.pdf> |