HW1: Experiment on Combing WS+BA Model to Generate A Social Network Graph

C44021032

Introduction:
The goal is to propose a model that can generate a realistic social network.

1. DESCRIPTION

A social network has 3 main properties: (1)short average path length, (2)high clustering coefficient, and (3)Power-law degree distribution.

Because Barabasi-Alber model and Watts-Strogatz model can satisfy two of properties mentioned above, the main idea is to combine these models to reach our goal. By extracting the concept in rewriting some of edges in WS model(decreasing average path length) and the probability that is proportional to the degrees in BA model when choosing a node to create links(make rich become richer), I hope the proposed method will remain properties(1) and (2), but also achieve (3).

The proposed model named **WSBA**, which is an abbreviation of WS model and BA model. In real world, people tend to being friends around them but still have the smaller chance to become friends with alienated people. Once a rewriting occurs, I introduce the concept of weighted probability when choosing the replacing nodes because when being friends with the people far away, it's easier to know the people with more friends(edges), which is the same idea in BA model.

2. SIMULATION COMPARISON

N: size of the network
K: average degree
p: rewrite probability
APL: average path length in graph
CC: cluster coefficient

WSBA model graph				
N	K	p	APL	CC
7673	14	0.15	4.14	0.27
282	7	0.15	3.73	0.22
7673	14	0.01	7.44	0.65
282	7	0.01	7.95	0.57

WS model graph				
N	K	p	APL	CC
7673	14	0.15	4.59	0.43
282	7	0.15	4.30	0.37
7673	14	0.01	9.51	0.67
282	7	0.01	11.76	0.58

2.1. power-law distribution issue

Fig 1, 2 show the WSBA doesn't fit power-law degree distrubution.

ALGORITHM 1: WSBA Model

Input: Graph size(N), average degree(K), and rewrite probability(p)
Output: The graph (G_{WSBA}).
Construct a regular lattice ring G such that N_G = N and K_G = K;
for *each edge in G, (e_i, e_j)* **do**
    $rand$ = Random(0, 1);
    if *($rand < p$)* **then**
        delete (e_i, e_j);
        $list$ = [ ];
        for *each node k different from i and j* **do**
            $list$.append(k);
        end
        choose a node k in $list$ by the probability p_k = d_k / $\sum_{n \in list} \mathrm{d}_n$;
        add (e_i, e_k);
    end
end
return G;

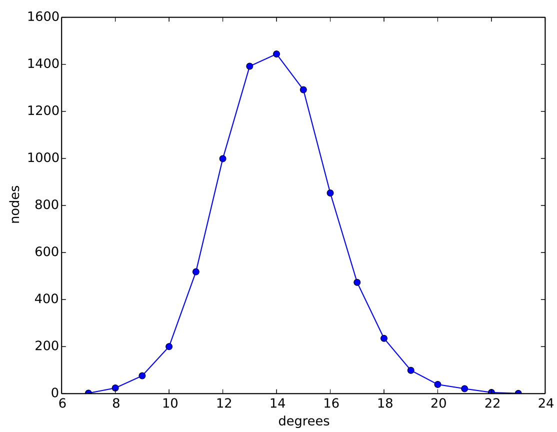

Fig. 1. WSBA degree distribution (N = 7673, K = 14, p = 0.15)

3. VISUALIZATION

Fig 3 is the graph generated by WSBA with N = 100, K = 6, p = 0.1, which is far away from the realistic social network graph because the difference of degree between nodes is not obvious.

Fig 4 is the graph generated by WSBA with N = 100, K = 6, p = 0.4, the difference of degree between nodes become more obvious but the cluster coefficient gets very low.

By increasing p, we can distinctly find two kinds of nodes: *with more edge* and *with less edge*, that is the property of the realistic graphs. Also, most of the nodes can reach each other within short path length, which is similar to the real world in social network. The different is that the CC of WSBA is lower than that of realistic graphs.

4. CONCLUSIONS

I found the WSBA and WS has some similar property. Both of their edges are not power-law distribution. Although introducing the weighted probability when rewriting, the distribution of degree in WSBA still approximately to Binomial distribution,

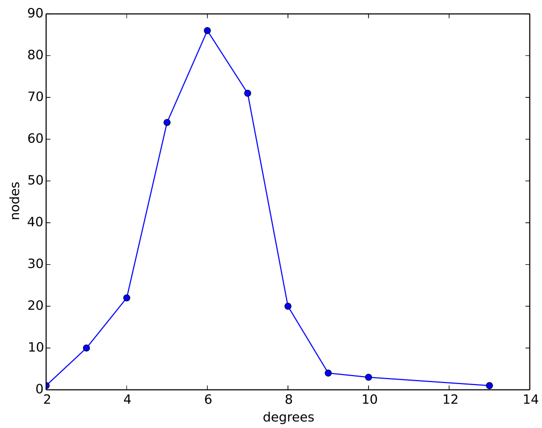

Fig. 2. WSBA degree distribution (N = 282, K = 7, p = 0.15)

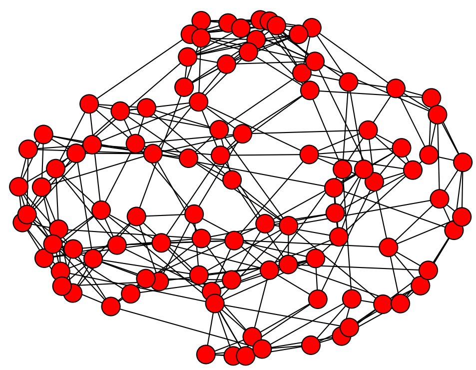

Fig. 3. WSBA graph (N = 100, K = 6, p = 0.1)

but is more bias and larger range. Increasing the rewriting probability will cause the decrease of the average path length and cluster coefficient, especially in WSBA.

As a result, WSBA performs better than WS in terms of APL and worse in terms of CC because it decreases more rapidly.

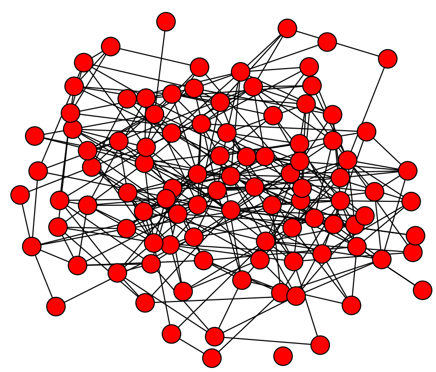

Fig. 4. WSBA graph (N = 100, K = 6, p = 0.4)