CASCADING BEHAVIOUR IN NETWORK ASSIGNMENT_REPORT
17BCS028

I have used karate.gml dataset for all over analysis
Considering the idea1 as all people are happy(blue nodes)
Considering the idea2 as the people who are sad(yellow nodes)

So we will induce idea2 in each test case and see how many people become sad

The main role in my code is of ***recreate function***,now what it does is:-

1. it take as parameters graph G,threshold
2. now we go one by one to every node (except those who adopted new idea2) and check
    a. M = number of neighbors of current node which have idea1
    b. N = number of neighbors of current node which have idea2
    c. p = N/(M+N)
    d. If p > threshold adopt new idea else retain the original idea

3.) return Graph the new g

Variation of input

NOTE* - this part of code will take lot of time to run since 34 nodes we willl take all combination.

1) Taking two new seeds at a time and run for 100 iteration(**since on running various input cases i found out ,that max cascading happen in uptill (or maye before)100 and graph repeats itself**, simple words it **achieve equilibrium**) each,This will give
    a) Cascade_size
    b) No. of iteration it took for cascading
    c) Whether complete or incomplete cascading
    d) No. of node which retain there characteristic

NOTE*- SInce above input case take a lot of time , so i also added for taking 5 seed node at random and 5 times so(5*5=25 combination)

2) Variation of seed nodes taking 5 node at random
    a) This will give
        i) Cascade_size
        ii) No. of iteration it took for cascading
        iii) Whether complete or incomplete cascading
        iv) No. of node which retain there characteristic
    b) Sample Output for this case

+ 5/34 vertices, from d1f8bed:
[1] 17 29 14 13 8
[1] 3
[1] "cascadesize"

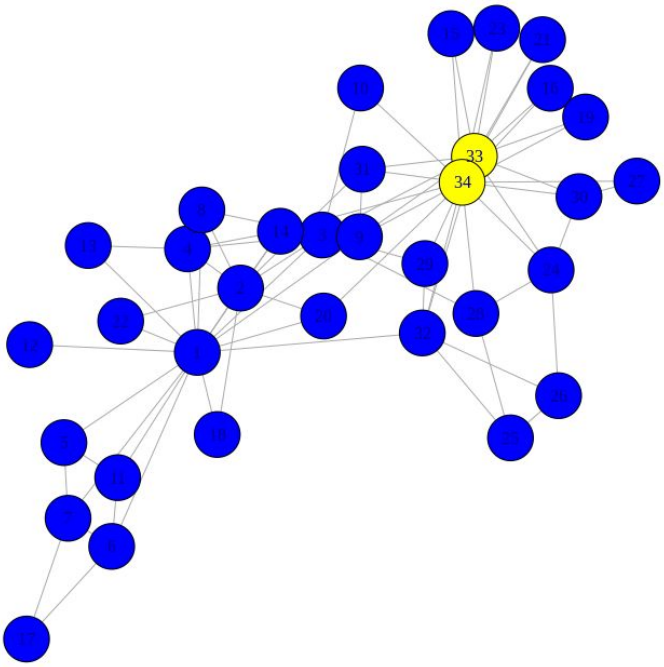
[1] 34
[1] "people who didn't adopt new idea"
[1] 0
[1] "complete"
+ 5/34 vertices, from d1f8bed:
[1] 1 30 8 19 31
[1] 2
[1] "cascadesize"
[1] 34
[1] "people who didn't adopt new idea"
[1] 0
[1] "complete"
+ 5/34 vertices, from d1f8bed:
[1] 26 18 27 23 20
[1] 100
[1] "cascadesize"
[1] 8
[1] "people who didn't adopt new idea"
[1] 26
[1] "incomplete"
+ 5/34 vertices, from d1f8bed:
[1] 26 31 27 18 9
[1] 100
[1] "cascadesize"
[1] 8
[1] "people who didn't adopt new idea"
[1] 26
[1] "incomplete"
+ 5/34 vertices, from d1f8bed:
[1] 5 7 31 27 26
[1] 4
[1] "cascadesize"
[1] 34
[1] "people who didn't adopt new idea"
[1] 0
[1] "complete"

3) **Varying threshold**
    a) Threshold = 0.6
    b) checking for each threshold and decrementing it 5 times
    c) This will give
        i) No. of iteration it took for cascading
        ii) Whether complete or incomplete cascading and corresponding size
        iii) No. of node which retain there characteristic

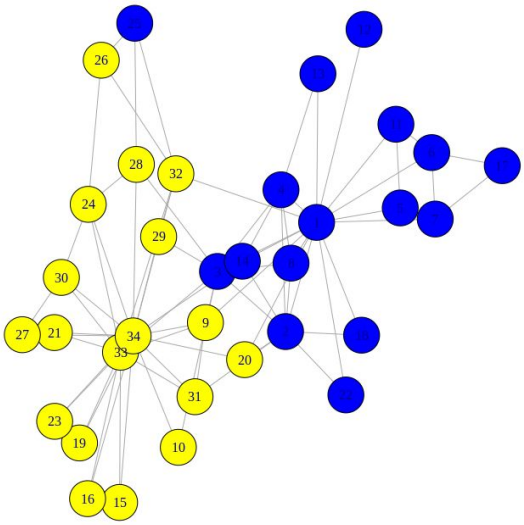
[1] 100
[1] "threshold"
[1] 0.5
[1] "cascadesize"
[1] 29
[1] "incomplete"
[1] 100
[1] "threshold"
[1] 0.4
[1] "cascadesize"
[1] 29
[1] "incomplete"
[1] 2
[1] "threshold"
[1] 0.3
[1] "cascadesize"
[1] 34
[1] "complete"
[1] 2
[1] "threshold"
[1] 0.2
[1] "cascadesize"
[1] 34
[1] "complete"
[1] 2
[1] "threshold"
[1] 0.1
[1] "cascadesize"
[1] 34
[1] "complete"

4) Run simple complete cascade case
#you can run code at start which have seed node(33,34) and threshold 0.3 for 4 times for below plots generation

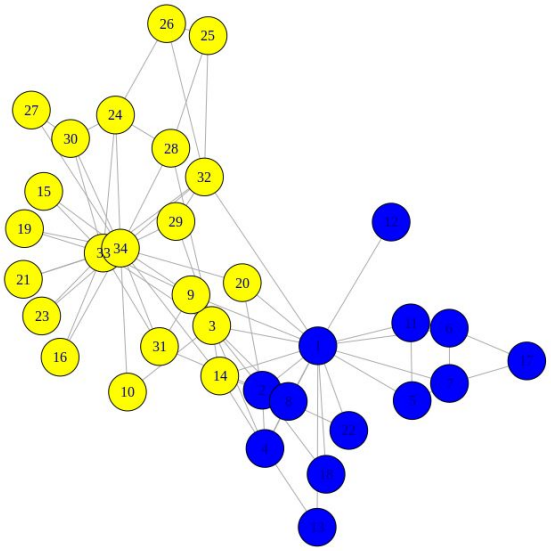
1)



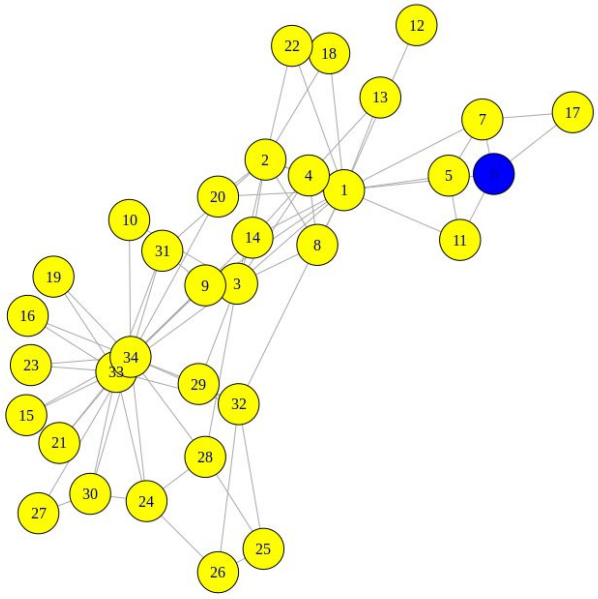
2)



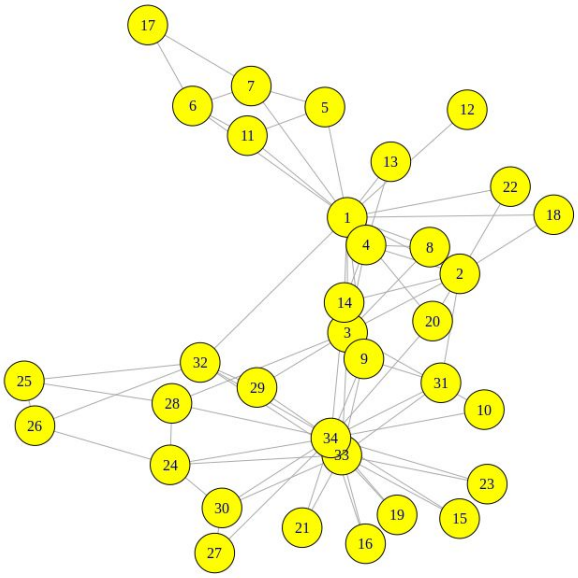
3)


26
25
27
24
30
28
32
15
29
19
33
34
21
20
9
23
3
31
16
14
10

4)


12
22
18
13
7
17
2
4
5
1
20
10
14
11
31
8
19
9
3
16
34
23
33
29
32
15
21
28
30
24
27
25
26

5)


17
7
6
5
12
11
13
22
18
1
4
8
2
14
20
3
9
32
31
25
29
28
10
26
34
33
24
23
30
19
15
21
16
27