INFX 576: Problem Set 8 - Network Processes*

Harkar Talwar

Due: Friday, May 25, 2018

Collaborators: Prateek Tripathi, Aakash Agrawal

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset8.Rmd` file from Canvas.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps8.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(statnet)
```

Problem 1: Network Diffusion

The following is a simple function to simulate cascading behavior in a social network. Write detailed comments for this code to demonstrate you understand the function. Feel free to modify or adjust the functionality.

```
# Set a seed value to ensure reproducibility of results
s = 761
set.seed(s)
```

```
# Function to simulate cascading behavior in a social network.
# Arguments:
# g: a network object, which when provided is used for analysis, rather than
#    generating a new random graph
# coords: coordinates of the vertices for the provided graph g
# network_size: the number of vertices
# network_density: the edge probability
# b_payout: the payout for adoption of behavior B
# a_payout: the payout for adoption of behavior A
# num_seeds: the number of initial adopters
```

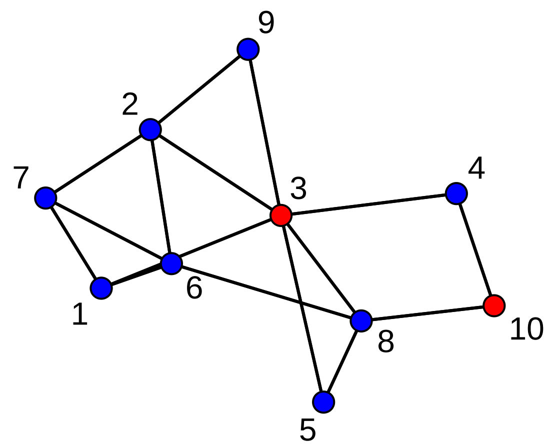
*Problems originally written by C.T. Butts (2009)

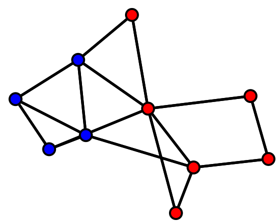
```r
# max_steps: max number of steps to allow for diffusion
diffusion <- function(network_size=10, network_density=0.2,
                      b_payout=2, a_payout=3, num_seeds=2, max_steps=10) {
  # create the initial network using a random graph,
  # with edge probability = network_density
  g <- network(rgraph(network_size, tprob=network_density), directed=FALSE)
  # set initial node colors to blue, indicating behavior B
  vertex_colors <- rep("blue", network_size)
  # sample initial adopters = num_seeds
  initial_nodes <- sample(1:network_size, num_seeds)
  # set the initial adopters to color red
  vertex_colors[initial_nodes] <- "red"
  # plot the initial state of the network with node colors depicting behavior
  coords = gplot(g, usearrows=FALSE, vertex.col=vertex_colors, displaylabel=TRUE)
  # compute the threshold used to evaluate if a node will switch behavior,
  # the value q indicates the proportion of a nodes's neighbors that need to have
  # adopted behavior A, so that the node can adopt A and maximize the payoff
  q = b_payout / (b_payout + a_payout)
  # partition the plotting area, to allow for upto 6*2 = 12 plots
  par(mar=c(0,0,0,0), mfrow=c(3,2))
  # create a copy of the colors depicting behavior adoption in the social network
  vertex_colors_new = vertex_colors
  step = 1

  # Repeat till max number of steps have not been reached or till all nodes have
  # adopted the new behavior, which ever occurs first
  while(step < max_steps && sum(vertex_colors=="red") < network_size) {

    # Repeat to test adoption of new behavior for each node in the current step
    for (i in 1:network_size) {
      # Fetch the neighbors of node i
      neighborInds <- get.neighborhood(g, i, type="combined")
      # Count the number of neighbors with behavior A, i.e. color red
      obsA <- sum(vertex_colors[neighborInds]=="red")

      # Check if the proportion of neighbors adopting A is >= to the threshold
      if (obsA/length(neighborInds) >= q) {
        # Set the color of this node to red, indicating adoption of behavior A,
        # however store the color in the copy of colors vector, so that the change
        # in color is not used in the current stage/step of the diffusion process
        vertex_colors_new[i] <- "red"
      }
    }
    # Update the colors to reflect the adoptions in the previous step
    vertex_colors = vertex_colors_new
    # Plot the network with the updated adoption status
    gplot(g, usearrows=FALSE, vertex.col=vertex_colors, coord=coords)
    Sys.sleep(.2)
    # Increment the step count before starting the next step of the diffusion
    step <- step + 1
  }
}
```

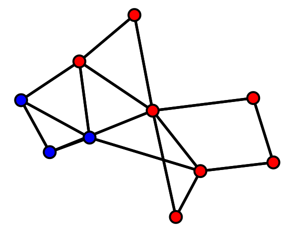
Experiment with this function to answer the following questions:

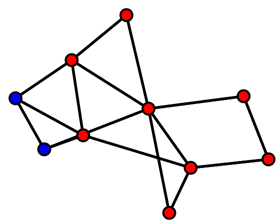
Configuration 1: Default Values

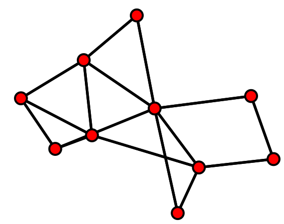
```
# network_size=10, network_density=0.2, b_payout=2, a_payout=3, num_seeds=2,
# max_steps=10
diffusion()
```





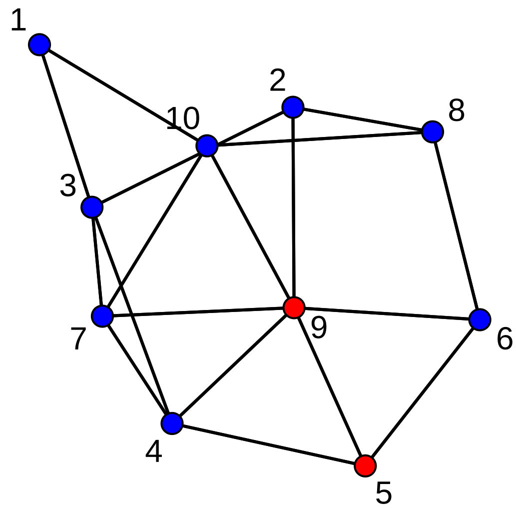


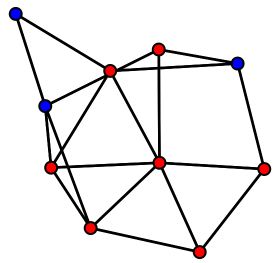


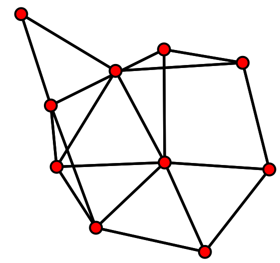


Configuration 2: High payoff on new behavior (a=10, b=2)

```
diffusion(network_size=10, network_density=0.2, b_payout=2, a_payout=10,
          num_seeds=2, max_steps=10)
```

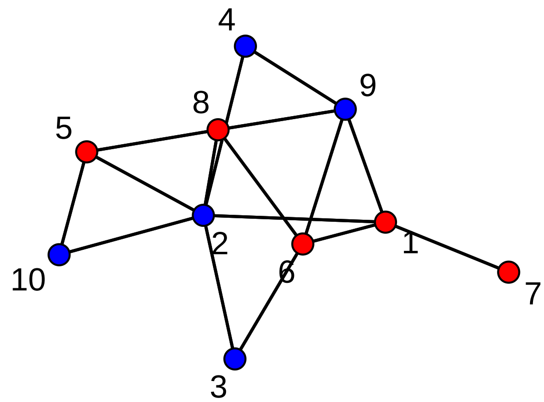

1
2
8
10
3
7
9
6
4
5

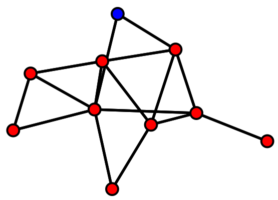


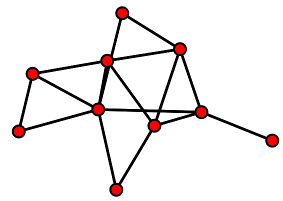


Configuration 3: Higher number of initial seeds (num_seeds = 5)

```
diffusion(network_size=10, network_density=0.2, b_payout=2, a_payout=3,
          num_seeds=5, max_steps=10)
```

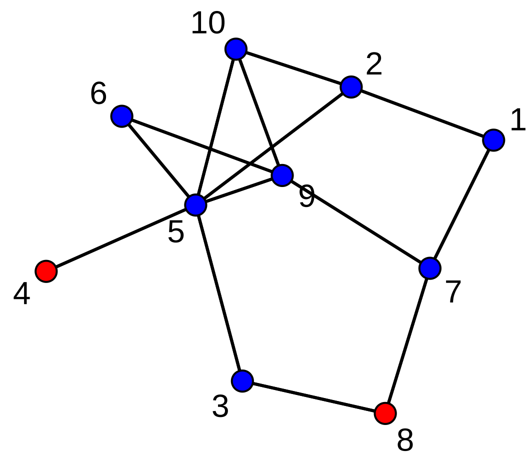

4
9
8
5
2
1
6
10
7
3

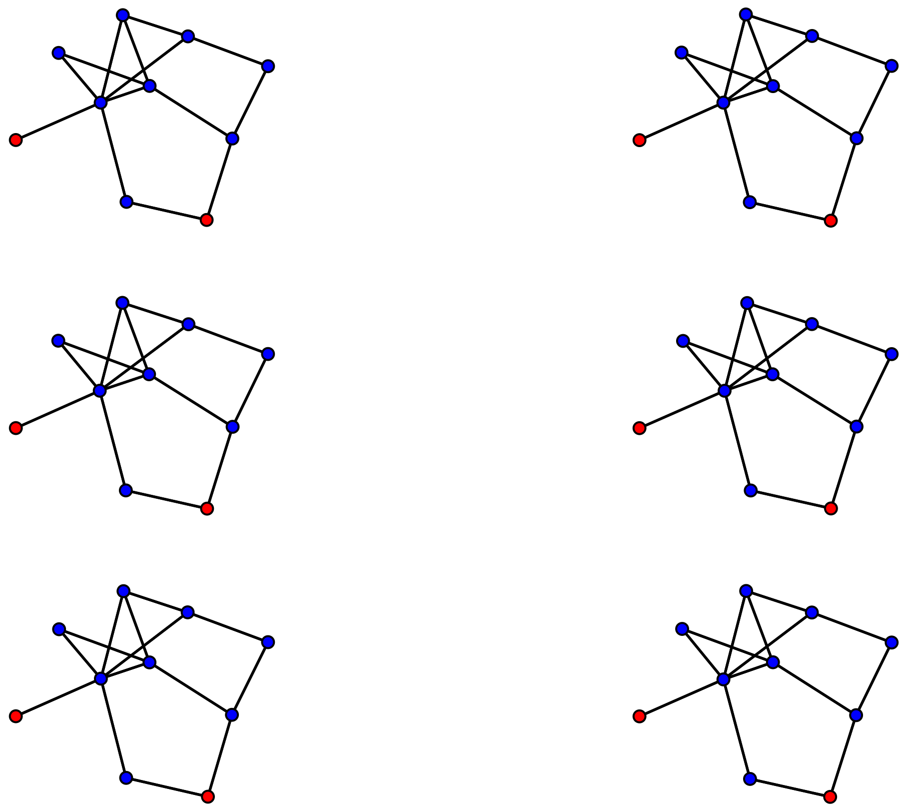


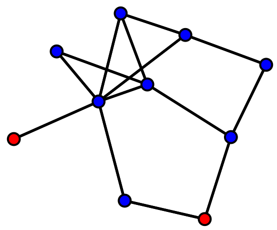


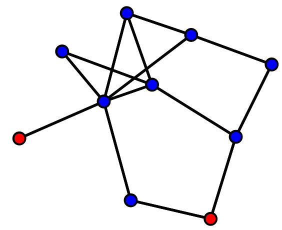
Configuration 4: High payoff on existing behavior (a=1, b=10)

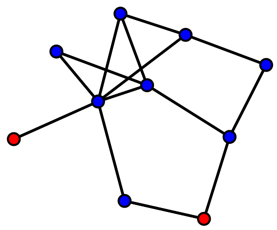
```
diffusion(network_size=10, network_density=0.2, b_payout=10, a_payout=1,
          num_seeds=2, max_steps=10)
```


10
2
6
1
9
5
7
4
3
8





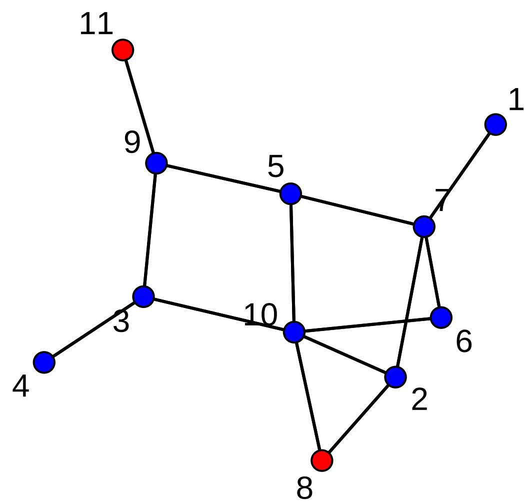


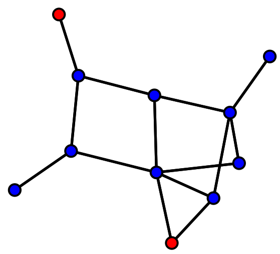
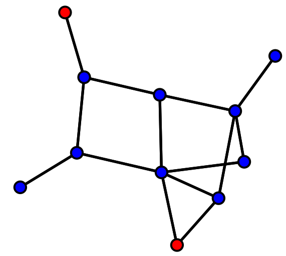
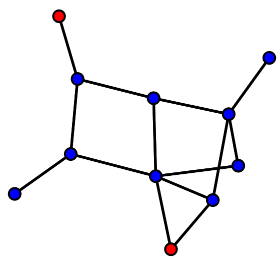
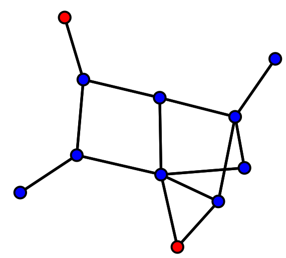
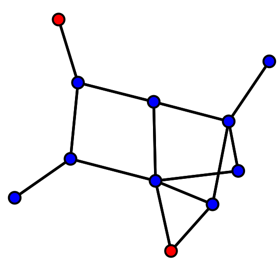
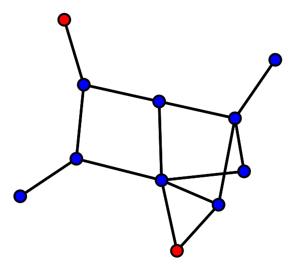


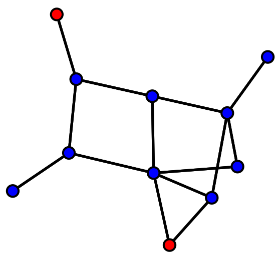
```r
# Set a different seed value to observe cluster density effect
s = 587
set.seed(s)
```

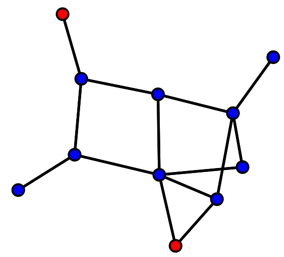
Configuration 5: Clusters of density (2/3) > 1-q (i.e. 3/5)

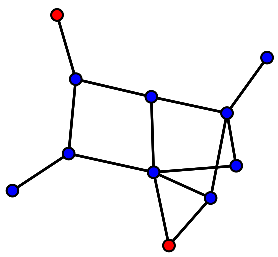
```r
diffusion(network_size=11, network_density=0.135, b_payout=2, a_payout=3,
          num_seeds=2, max_steps=10)
```


11
1
9
5
7
10
3
6
4
2
8







When does this simulation result in everyone switching to A?

1. As seen in Configuration 1, the adoption of behavior A depends on the threshold value computed as $\frac{b}{a+b}$, where b, a are payouts for behaviors B, A respectively. For a given node, if the proportion of neighbors adopting A is greater than or equal to the threshold, then the node would adopt A.

2. The spread therefore depends on a number of factors, and the following conditions, alone or in tandem can result in everyone switching to A:

- High payoff (a=10) for the adoption of new behavior (Configuration 2 above). This reduces the threshold to $\frac{2}{2+10} = 0.16$, causing everyone to adopt A in just 2 steps.
- Higher number of initial adopters (Configuration 3 above). Here the number of initial adopters was increased from 2 to 5, leading to a quicker spread of the behavior (complete cascade in two steps). A higher number of initial adopters alone however may not lead to complete adoption, since the network structure and thresholds have an important role to play.
- Absence of tightly knit communities and homophily. As we will discuss in the question below, tightly knit communities can offer severe resistance to the spread of new behaviors. For configurations 1, 2 and 3, the remaining network (apart from the seeds) does not contain clusters of density greater than 1-q, q being the threshold. This indicates the absence of tightly knit groups due to which these configurations result in a complete cascade.

What causes the spread of A to stop?

Configuration 4 and 5 above illustrate scenarios which can cause the spread of behavior A to stop.

- In configuration 4, we see that the pay-off (a=1) on the new behavior A is far lower than that on the existing behavior B (b=10). This causes the innovation/behavior to not spread beyond the initial adopters.

- Further, in configuration 5 we see that nodes 3,5,9,10 form a cluster of density 2/3 = 0.66 which is greater than 1-q = 1-2/5 = 0.6. As we know, if the remaining network (apart from the seeds) contains a cluster of density greater than 1-q, then the set of initial adopters will not cause a complete cascade. The nodes 3,5,9,10 are thus part of a tightly knit community, preventing complete adoption.

- Lastly, configuration 5 shows us that the position of the initial adopters in the network also plays a role in the spread of behaviors. Had the initial adopters been nodes 8 and 10, instead of 8 and 11, there would have at least been some spread of the behavior beyond the initial adopters.