CS 5854: Networks, Crowds, and Markets
Homework 2

Instructor: Rafael Pass   TAs: Cody Freitag, Drishti Wali

Assigned: October 2, 2019   Due: October 27, 2019, 11:59 pm

Ananya Shivaditya aks298

Collaborators: I worked by myself.
Outside resources: I used the following outside resources:

1. http://www.cs.cornell.edu/~rafael/networks.pdf

Late days: I have used 2 late days on this assignment.

Part 1: Best-Response Dynamics

1. (a) Let G_1 and G_2 be games with the same set of players and action sets for each player, and *assume that BRD converges to a PNE* in both of these games. Consider a game $G = (G_1 + G_2)$ where each player plays a single action for both G_1 and G_2 (i.e. plays the same action in both games at the same time) and receives utility equal to the *sum* of the utilities they would earn from G_1 and G_2. *Will BRD converge in this game?* If so, prove it; if not, find a counterexample and argue why it doesn't converge. (*Hint:* Start with a game that may not converge and try to split it into two separate games.)

   (b) Assume in a particular game $G = G_1 + G_2$ that BRD does converge. Will it necessarily converge to a state that is also a PNE in either G_1 or G_2? If so, prove it; if not, find a counterexample.

Solution:

(a) No - will not necessarily converge. Example:

$$G = (G_1 + G_2)$$

G	$(*, X)$	$(*, Y)$
$(X, *)$	(0, 1)	(1, 0)
$(Y, *)$	(1, 0)	(0, 1)

G_1	$(*, X)$	$(*, Y)$
$(X, *)$	(0, 0)	(1, 0)
$(Y, *)$	(1, 0)	(0, 0)

G_2	$(*, X)$	$(*, Y)$
$(X, *)$	(0, 1)	(0, 0)
$(Y, *)$	(0, 0)	(0, 1)

BRD will not converge in G. While it will converge in G_1 and in G_2

(b) Yes, BRD will converge to a state that is also a PNE in either G_1 or G_2. Since G is known to have BRD converge, this must mean that the potential reaches a maximum beyond which no more deviation is possible. If we use the same function for potential on G_1 and G_2, we know that it will be maximum again at a state of BRD convergence for each game. Given that $G = G_1 + G_2$, this must mean the PNE in G is the sum of individual potentials of G_1 and G_2. Hence the PNE in G which is the state with maximum potential must be the same as either in G_1 or G_2.

$\square$

2. Consider the process of "better-response dynamics" rather than BRD, where instead of choosing a player's best response (i.e. the response that maximizes their utility given others' actions), a player chooses any response that *strictly* increases their utility given other players' actions.

   (a) Does the process of "better-response dynamics" still converge in games for which there is an ordinal potential function Φ? Prove it or show a counterexample. (*Hint:* This is in the notes now. It suffices to reference the correct theorem.)

   (b) Can you think of a game for which BRD converges but "better-response dynamics" might not? Show an example or justify that one doesn't exist. (*Hint:* Remember that the better-response dynamics graph can have edges that the BRD graph might not. What if those edges formed a loop?)

* **Bonus Question 1.** Recall the Traveler's Dilemma that we studied in chapter 1. We have already illustrated why BRD converges in this game; find a **weakly** ordinal potential function Φ over states of this game and use it to definitively prove the convergence of BRD. Make sure to justify that the potential function you find is weakly ordinal (you can do this via computer if you wish).

* **Bonus Question 1.5.** (*This may be very difficult!*) Determine whether *better-response dynamics* converge for the Traveler's Dilemma. No formal proof is necessary, and you need not come up with a potential function. You are free to either use simulations, show (experimentally) that the graph contains no cycle, or find an ordinal potential function and either prove it or experimentally verify.

Solution:

(a) Yes, BRD does converge. Starting from an arbitrary state, begin running BRD. The potential will increase every time a player better responds and there are only a finite number of states in the game. So, after n number of better - response steps, potential will be maximal and further deviation will not be possible.

(b) Game G has BRD convergence but not better response dynamics convergence.

G	$(*, X)$	$(*, Y)$	$(*, Z)$
$(X, *)$	(3, 3)	(1, 0)	(0, 1)
$(Y, *)$	(0, 1)	(0, 0)	(1, 0)
$(Z, *)$	(1, 0)	(0, 1)	(3, 3)

Bonus 1

Bonus 1.5

□

Part 2: Networked Coordination Games

3. Consider the following simple coordination game between two players:

	$(*, X)$	$(*, Y)$
$(X, *)$	(x, x)	(0, 0)
$(Y, *)$	(0, 0)	(y, y)

Show how we can pick x and y and then modify this payoff matrix by adding an intrinsic utility for a single player and a single choice (e.g. give player 1 some intrinsic utility for choice Y) such that the socially optimal state of the game is no longer an equilibrium.

Solution: Let x = 4 and y = 1. Now let state (Y, X) be (5, 0) i.e. Player 1 is given intrinsic utility 5 for choosing Y. This changes the playoff matrix to:

	$(*, X)$	$(*, Y)$
$(X, *)$	(4, 4)	(0, 0)
$(Y, *)$	(5, 0)	(6, 1)

Here we see that the socially optimal state(X, X) is no longer an equilibrium, as player 1 would deviate to Y and gain utility. BRD will converge at (Y,Y) which gives utility of $(6, 1)$ and social utility of 7. Max social utility which was at (X, X) was 8, but that's not in equilibrium anymore. □

4. Consider a networked coordination game on a complete graph of five nodes. Assume for simplicity that all edges represent the same coordination game (that is, $Q_e(X,X) = Q_{e'}(X,X)$ for any pair of edges e, e', and respectively for Y, but it is not necessarily the case that $Q_e(X,X) = Q_e(Y,Y)$), and that all nodes have the same intrinsic values for X and Y (that is, $R_i(X) = R_j(X)$ for any nodes i, j, and respectively for Y).

   (a) Show a possible assignment of intrinsic and coordination utilities such that every pure-strategy Nash equilibrium of the resulting game must be socially optimal. Justify that this holds for your construction.

   (b) Is there a possible assignment of intrinsic and coordination utilities such that there exists a pure-strategy Nash equilibrium with social welfare less than half of the maximum attainable social welfare? If not, prove it. If so, show such an assignment and explain why your construction does not violate Theorem 4.5 from the notes (the price of stability). (*Hint:* Theorem 4.5 only talks about the *best* equilibrium. Try to make a simple two-node game with one very good equilibrium and one bad equilibrium, and then extend what you find there to the five-node graph. It also won't be necessary to use intrinsic values for this part.)

Solution:

(a) Let every player's intrinsic utility be 1, for both states X and Y. Let coordination utilities be 1, for both states X and Y for every player. Therefore, the 2 PNEs for this game would be if all players chose the same state, either X or Y. This is also the social optimum, with highest social welfare as 5*(1+1) = 10.

(b) Yes, it is possible for an assignment of intrinsic and coordination utilities to have a PNE with social welfare less than half of the maximum attainable social welfare. Let the playoff matrix for any two players of the five, be the following, where we set x = 10, y = 1.

	$(*, X)$	$(*, Y)$
$(X, *)$	(10, 10)	(0, 0)
$(Y, *)$	(0, 0)	(1, 1)

Now set intrinsic value of choosing Y for player 1 be 2. And set intrinsic value of choosing X for player 2 be 2. So the playoff matrix becomes:

	$(*, X)$	$(*, Y)$
$(X, *)$	(10, 10)	(0, 2)
$(Y, *)$	(2, 0)	(3, 3)

Here we see that the maximum attainable social welfare is 5*(10 = 50. But the PNE where every player chooses Y has social welfare 5*(3) = 15. This is less than half the maximum attainable social welfare.

This construction does not invalidate Theorem 4.5 since the theorem only talks about the best PNE. Here, we are asked the social welfare of any PNE, to be less than half the maximum attainable. So, no invalidation of the theorem has occurred.

□

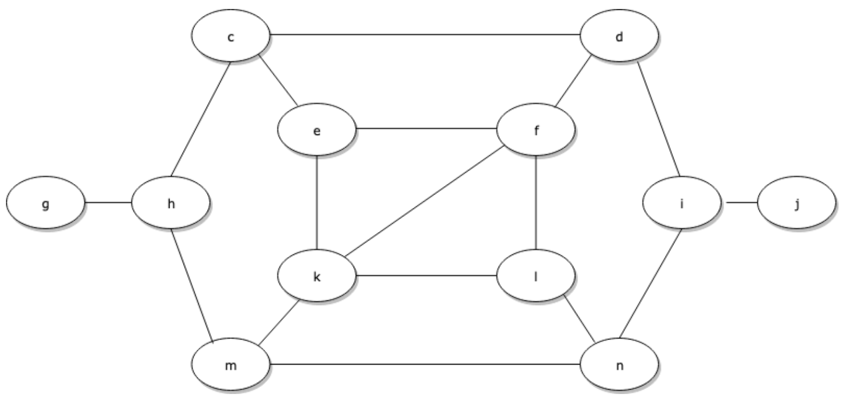
Part 3: Cascading Behavior in Networks

5. Consider the network in Figure ??. Suppose that each node starts with the behavior Y, and has a $q = 2/5$ threshold for adopting behavior X. (That is, if at least 2/5 of a node's neighbors have adopted X, that node will as well.)

   (a) Let e and f form a set S of initial adopters of X. Which nodes will eventually switch to X? (Assume that S will not participate in BRD.)

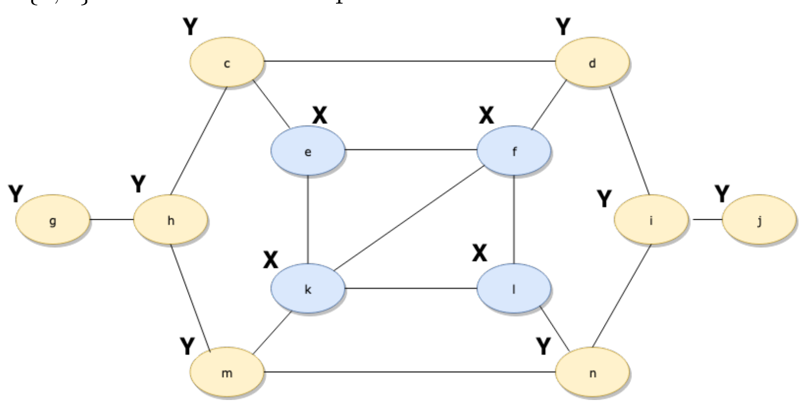
   (b) Find a cluster of density $1 - q = 3/5$ in the part of the graph outside S which blocks X from spreading to all nodes starting from S.

   (c) Add one node to S such that a complete cascade will occur at the threshold $q = 2/5$. Demonstrate how the complete cascade could occur (i.e. in what order nodes will switch).



Solution:

(a) S = {e, f} makes k and l adopt X too.



(b) Cluster of density 1 - q = $\frac{3}{5}$ in the part of the graph outside S which blocks X from spreading to all nodes starting from S includes {g, h, c, d, i, j, n, m }

(c) Including h in S will make X cascade across the entire network. New S = {h, e, f}
Complete cascade could occur in the following order:
k − > l − > c − > g − > m − > n − > d − > i − > j

□

6. (a) Formulate a graph G, threshold q, and set S of initial adopters such that, assuming we start with S choosing X (and, importantly, able to participate in BRD) and other nodes choosing Y, we can either end up with every node in G playing X or with every node in G playing Y, depending on the order of switches in the BRD process.

   (b) What must be true of the set density of S for the above property to hold?

   (c) What must be true of the set density in any subset of $V \setminus S$ for the above property to hold?

Solution:

(a) Let G be fully connected graph with 4 nodes = {A, B, C, D}, with threshold $q = \frac{1}{3}$ and S = {A} If order of switching in BRD (starting from A in S adopting X) is
A(X) $->$ B(X) $->$ C(X) $->$ D(X) then end state will have all payers choosing X.
Else, if order of switching in BRD (starting from A in S adopting X) is
A(X) $->$ B(X) $->$ A(Y) $->$ B(Y) then end state will have all players choosing Y.

(b) Set density of S must be equal to threshold i.e. in this case, 1/3.

(c) Any subset of S must have set density less than threshold.

$\square$

Part 4: Traffic Networks

7. There are two cities, A and B, joined by two routes which pass through towns C and D respectively. There are 120 travelers who begin in city A and must travel to city B, and may take the following roads:

   - a local street from A to C with travel time $15 + x$, where x is the number of travelers using it,
   - a highway from C to B with travel time 90,
   - a highway from A to D with travel time 90, and
   - a local street from D to B with travel time $15 + y$, where y is the number of travelers using it.

   (a) Draw the network described above and label the edges with their respective travel times. The network should be a directed graph (assume that all roads are one-way). Note: you may just describe the graph if you are typing the assignment up.

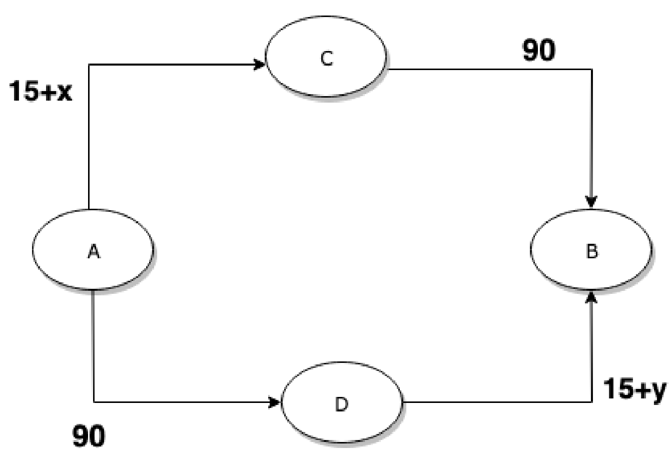
   (b) Find the Nash equilibrium values of x and y. (Show that this is the equilibrium.)

   (c) The government now adds a new *two-way* road connecting the nodes where local streets and highways meet. This new road is extremely efficient and requires no travel time. Find the new Nash equilibrium.

   (d) What happens to the total travel time as a result of the availability of the new road? (You don't need to explain, a calculation is fine.)

   (e) Now suppose that the government, instead of closing the new road, decides to assign routes to travelers to shorten the total travel time. Find the assignment that minimizes the total travel time, and determine the total travel time using this assignment. (*Hint:* It is possible to achieve a total travel time less than the original equilibrium. Remember that, with the new road, there are now four possible routes that each traveler can take.)

Solution:



(a)

(b) x = y = 60. This is PNE because travel time across both paths is equal, i.e. 90+15+60 = 165. No person would deviate since their travel time would increase if they did.

(c) New PNE is when x = y = 75.

(d) This increases each person's travel time to 180, but is still PNE since no one would deviate from this state.

(e) Total travel time is minimized when :

- Path AC is taken by 38 people.
- Path AD is taken by the remaining 82 people.
- Path DC is taken by 45 people.
- Path CB is taken by 83 people.
- Path DB is taken by the remaining 37 people.

OR when:

- Path AC is taken by 37 people.
- Path AD is taken by the remaining 83 people.
- Path DC is taken by 45 people.
- Path CB is taken by 82 people.
- Path DB is taken by the remaining 38 people.

□

Part 5: Experimental Evaluations

8. This problem will make use of the data set available at:

   [http://snap.stanford.edu/data/egonets-Facebook.html].

   In particular, please refer to the file "facebook_combined.txt.gz"; it contains a text file listing all 88,234 edges (*undirected* edges representing Facebook friendship) in a sampled 4,039-node network (nodes are numbered 0 to 4038). It will be useful, for problem 8 to be able to input a graph presented in such a format into your code.

   (a) Consider the contagion examples that we observed in chapter 5 of the notes. Given an *undirected* graph, a set of early adopters S, and a threshold q (such that a certain choice X will spread to a node if more than q fraction of its neighbors are playing it), produce an algorithm that permanently infects the set S of early adopters with X and then runs BRD on the remaining nodes to determine whether, and to what extent, the choice will cascade through the network. (*Note:* "BRD" in this case is simply the process of iteratively deciding whether there is a node that will switch its choice and performing this switch.)

   Once again, turn in your code and verification that your algorithm works on a few simple test cases. In particular, include the output on the two examples from Figure 4.1 in the notes; let S be the set of nodes choosing X in the figure, and, for each of the two graphs, include one example with a complete cascade and one without one (and specify what value of q you used for each).

   (b) Run your algorithm several (100) times on a fairly small random set of early adopters ($k = 10$) with a low threshold ($q = 0.1$) on the Facebook data set. What happens? Is there a complete cascade? If not, how many nodes end up being "infected" on average?

   (c) Run your algorithm several (10) times with different values of q (try increments of 0.05 from 0 to 0.5), and with different values of k (try increments of 10 from 0 to 250). Observe and record the rates of "infection" under various conditions. What conditions on k and q are likely to produce a complete cascade in this particular graph, given your observations?

   * **Bonus Question 2.** (Optional, extra credit awarded depending on quality of solution.) Design an algorithm that, given a graph and a threshold q, finds (an approximation of) the smallest possible set of early adopters that will cause a complete cascade. Try running it on the Facebook data with different values of q and seeing how large a set we need.

   Solution:

   (a) Answer given below.

```
[41] # testing 8 a) with figure 4.1
     def test_4_1():
         G1 = {0: [1], 1: [0,2], 2: [1,3], 3:[2]}
         S1 = [0, 1]
         # complete cascade in left figure 4.1 can occur for any q <= 0.5
         print('Output of complete cascade:  ', cascade(G1,S1,0.49)) # using value of q as 0.49
         # incomplete cascade in left figure 4.1 can occur for any q > 0.5
         print('Output of incomplete cascade:  ', cascade(G1,S1,0.51)) # using value of q as 0.51
         G2 = {0: [1], 1: [0,2,3], 2: [1], 3:[1, 4, 5], 4:[3], 5:[3, 6], 6: [5]}
         S2 = [0, 1, 2]
         # complete cascade in right figure 4.1 can occur for any q <= 1/3
         q = 1/3
         print('Output of complete cascade:  ', cascade(G2,S2,q)) # using value of q as 1/3
         # incomplete cascade in right figure 4.1 can occur for any q > 1/3
         print('Output of incomplete cascade:  ', cascade(G2,S2,0.5)) # using value of q as 0.5
```

```
[45] test_4_1()
```

```
Output of complete cascade:   {0: 'X', 1: 'X', 2: 'X', 3: 'X'}
Output of incomplete cascade:   {0: 'X', 1: 'X', 2: 'Y', 3: 'Y'}
Output of complete cascade:   {0: 'X', 1: 'X', 2: 'X', 3: 'X', 4: 'X', 5: 'X', 6: 'X'}
Output of incomplete cascade:   {0: 'X', 1: 'X', 2: 'X', 3: 'Y', 4: 'Y', 5: 'Y', 6: 'Y'}
```

```
# ans 8 b)
def fb_data():
    G = make_graph()
    infected = []
    yes = 0
    q = 0.1
    for i in range(100):
        S = np.random.randint(0,4039,10)
        result = contagion_brd(G,S,q)
        inf = sum(result[x]=='X' for x in result)
        infected.append(inf)
        if inf == len(result):
            yes += 1
    if yes > 0:
        print('There is (are) {} complete cascade(s)'.format(yes))
    print('On average, {} nodes are infected'.format(sum(infected)/100))
```

```
[58] fb_data()
```

```
There is (are) 32 complete cascade(s)
On average, 3434.33 nodes are infected
```

(b)

(c) Only the values of 0 for q give complete cascades. Code attached separately.

□

9. Consider the following problem: There are n Uber drivers and m potential riders. At a fixed point in time, each driver has a list of compatible riders that she can pick up. Our goal will be to match drivers to riders such that the most riders at this time are picked up. We will use the maximum-flow algorithm, described in Chapter 6 of the notes to do this.

(a) First, implement an algorithm that, given a *directed* graph, a source s, a sink t, and edge capacities over each edge in E, computes the maximum flow from s to t (you must implement this algorithm yourself). Turn in your code and verify that your algorithm works on a few simple test cases. In particular, test your algorithm on the graphs in Figures 6.1 and 6.3 from the lecture notes and submit the output.

(b) Given a set of n drivers, m riders, and sets of possible riders that each driver can pick up,

i. explain how we can use this maximum-flow algorithm to determine the the maximum *number* of matches, and

ii. explain how we can additionally extend this to actually find the matchings.

(*Hint:* See the notes if you are confused about how to do this.)

(c) Implement a maximal matching algorithm for Uber drivers and riders. Specifically, given n drivers with constraints specified on m riders, computed the assignments of drivers to riders. Test your algorithm on at least 2 examples (with at least 5 riders and drivers each). Explain your examples and your results.

(d) Now consider the case where there are n drivers and n riders, and the drivers each driver is connected to each rider with probability p. Fix $n = 1000$ (or maybe 100 if that's too much), and estimate the probability that all n riders will get matched for varying values of p. Plot your results.

* **Bonus Question 3:** In the above example, let $p^*(n)$ be the smallest value of p where all n riders get matched with at least 99% probability. Prove bounds on what $p^*(n)$ is as a function of n, e.g. is $p^*(n) \geq 1/n$ or $p^*(n) \leq 1/2$. You will get partial credit for providing experimental evidence towards a proposed idea.

Solution:

(a) **Output for Figure 6.1:**
Output for Figure 6.3:

(b)

(c)

(d)

□