CSE 6240 Spring 2020 - Homework #3 Analyzing Graphs

Due: Apr 13, 2020

The aim of this homework is to give hands-on experience with Information-based Cascading.

Guidelines:

❖ You are given a solution template (.ipynb file) and you are required to add your code at the appropriate places and submit it.
❖ The solution template contains questions with multiple sub-parts for each one.
❖ For the coding exercises, please add your code below the comment "Add your code here."
❖ For the theory questions, please add your answers in a text cell below the questions. The cell type can be changed under Cell -> Cell Type.
❖ Points distribution for each question is added for your reference.
❖ Please **do not** modify any of the function definitions or documentation, nor add any other libraries or custom functions, apart from the ones imported for you.
❖ All the deliverables for the questions are mentioned at the end of the templates.

1. Decision-based Cascades: A Local Election

It's election season and two candidates, Candidate A and Candidate B, are in a hotly contested city council race in sunny New Suburb Town. You are a strategic advisor for Candidate A in charge of election forecasting and voter acquisition tactics.

Based on careful modeling, you've created two possible versions of the social graph of voters. Each graph has 10,000 nodes, where nodes are denoted by an integer ID between 0 and 9999. The edge lists of the graphs are provided in the homework bundle. Both graphs are **undirected**.

Given the hyper-partisan political climate of New Suburb Town, most voters have already made up their minds: 40% know they will vote for A, 40% know they will vote for B, and the remaining 20% are undecided. Each voter's support is determined by the last digit of their node id. If the last digit is 0–3, the node supports A. If the last digit is 4–7, the node supports B. And if the last digit is 8 or 9, the node is undecided.

The undecided voters will go through a 10-day decision period where they choose a candidate each day based on the majority of their friends. The **decision period** works as follows:

1. The graphs are initialized with every voter's initial state (A, B, or undecided).

2. In each iteration, every undecided voter decides on a candidate. Voters are processed in increasing order of node ID. For every undecided voter, if the majority of their friends support A, they now support A. If the majority of their friends support B, they now support B. "Majority" for A means that strictly more of their friends support A than the number of their friends supporting B, and vice versa for B (ignoring undecided friends).

3. If a voter has an equal number of friends supporting A and B, we assign support for A or B in alternating fashion, starting with A. In other words, as the voters are being processed in increasing order of node ID, the first tie leads to support for A, the second tie leads to support for B, the third for A, the fourth for B, and so on. This alternating assignment happens at a *global level* for the whole network, across all rounds. (Keep a single global variable that keeps track of whether the current alternating vote is A or B, and initialize it to A in the first round. Then as you iterate over nodes in order of increasing ID, whenever you assign a vote using this alternating variable, change its value afterwards.)

4. When processing the updates, use the values from the current iteration. For example, when updating the votes for node 10, you should use the updated votes for nodes 0–9 from the current iteration, and nodes 11 and onwards from the previous iteration.

5. There are 10 iterations of the process described above.

6. On the 11th day, it's election day, and the votes are counted.

Note that only the undecided voters go through the decision process. The decision process does not change the loyalties of those voters who have already made up their minds. Voters who are initially undecided may change their mind on each iteration of this process.

1.1 Basic Setup and Forecasting

Start your work with the starter code provided. Read in the two graphs and assign the initial vote configurations to the network. Then, perform the 10 iterations of the voting process. Which candidate wins in Graph 1, and by how many votes? Which candidate wins in Graph 2, and by how many votes?

For sanity check, neither of these two numbers (how many votes) is larger than 300.

1.2 TV Advertising

You have amassed a substantial war chest of $9000, and you have decided to spend this money by showing ads on the local news. Unfortunately, only 100 New Suburb Townians watch the local news—those with ids 3000–3099. However, your ads are extremely persuasive, so anyone who sees the ad is immediately swayed to vote for candidate A regardless of his/her previous decision. You may spend $1000 at a time on ads. The first $1,000 reaches voters 3000–3009, the second $1000 reaches voters 3010–3019, and so on. In other words, the total of $k in advertising would reach voters with ids from 3000 to 3000 + k/100 -1. This advertising happens before the decision period. *After voters are persuaded*

by your ads, they never change their minds again.

Simulate the effect of advertising spending on the two possible social graphs. First, read in the two graphs again and assign the initial configurations as before. Now, before the decision process, you purchase $k of ads and go through the decision process of counting votes.

For each of the two social graphs, plot $k (the amount you spend) on the x-axis (for values k = 1000, 2000, . . . , 9000) and the number of votes for A minus the number of votes for B on the y-axis. Put these on the same plot. What's the minimum amount you can spend to win the election in each of the two social graphs?

Note that the TV advertising affects all of the voters who see the ads and not just those who are undecided.

1.3 Wining and Dining the High Rollers

TV advertising is only one way to spend your campaign war chest. You have another idea to have a very classy $1000 per plate event for the high rollers of New Suburb Town (the people with the highest degree in the social graph). You invite high rollers in order of how many people they know, and everyone that comes to your dinner is instantly persuaded to vote for candidate A regardless of his/her previous decision. For each high roller you spend $1000 to this persuasion. This event will happen before the decision period. When there are ties between voters with the same degree, the high roller with lowest node ID gets chosen first.

Simulate the effect of the high roller dinner on the two graphs. First, read in the graphs and assign the initial configuration as before. Now, before the decision process, you spend $k on the fancy dinner and then go through the decision process of counting votes.

For each of the two social graphs, plot $k (the amount you spend) on the x-axis (for values k = 1000, 2000, . . . , 9000) and the number of votes you win by on the y-axis (that is, the number of votes for A less the number of votes for B). What's the minimum amount you can spend to win the election in each of the two social graphs?

Note that wining and dining sways all the voters to vote for A and not just those who are undecided.

1.4 Analysis

Plot the degree distributions on a log-log scale of the two graphs on the same plot. Although both graphs have roughly the same number of edges, degree distributions are actually very different. In 1–2

sentences, briefly summarize how this difference explains the results of Question 1.3.