Professor Nicole Eikmeier
CSC-395, Fall 2023

Project 3
Data and Code Explanation

Assigned: N/A
Due: N/A

The Simulation

In this simulation, we construct a social network, where each day, users can read news articles from a set of news sources and then potentially have the articles they read show up in the news feeds of other users in the social network. The network will contain some number of “leftleaning” and some number of “right-leaning” users. The number of each type of user can be set in the user interface of the simulation when it is run (with up to a maximum of 30 of each type of user). We consider all users (regardless of political leaning) to be friends in the network, so news articles read by any user may potentially appear in the news feed of any other user.

Note that if you want to modify the maximum number of each type of user (which is not required for this assignment), the limits can be changed by setting the constants `MAX_LEFT_LEANING_USERS` and `MAX_RIGHT_LEANING_USERS` in the file `constants.py`. There are also several other constants (all commented) in this file which can be changed, if you desire. But, again, it is not necessary to change any of these values to do this assignment. We just note it here in case you want to modify the simulation code.

News Sources

There are 10 news sources in the simulation. The news sources are considered to be on a political spectrum from “right” to “left” with regard to the articles/content that they produce. Such a set-up is not unrealistic, as (for example) the website AllSides.com, which provides ratings of political leanings for news websites, provides a categorization of news sources below:

Note that throughout the rest of the assignment description, we will use the term “article” to refer to a piece of content that comes from a news source. As far as the simulation is concerned, the actual content of an article does not matter. Only the news source that an article came from matters with respect to, for example, if the article is read by a user or not. Thus, the term “article” is really synonymous with the news source that the given article is from.

Left-leaning and right-leaning users

Left-leaning and right-leaning users in the social network have different true probabilities of reading articles from across the set of 10 news sources. The true probability denotes the chance that when a user is presented with an article from a given news source, the user will actually read that article.

The true probabilities (rounded to two decimal places) for "left-leaning" users for reading the 10 news sources are as follows (i.e., starting at 70% and decreasing linearly to 30%):

Source	1	2	3	4	5	6	7	8	9	10
P(read)	0.70	0.66	0.61	0.56	0.52	0.48	0.43	0.39	0.34	0.30

The true probabilities (rounded to two decimal places) for "right-leaning" users for reading the 10 news sources are as follows (i.e., starting at 30% and increasing linearly to 70%):

Source	1	2	3	4	5	6	7	8	9	10
P(read)	0.30	0.34	0.39	0.43	0.48	0.52	0.56	0.61	0.66	0.70

Model of individual users reading news

Each day in the simulation (the number of days the simulation lasts is a value that can be set through the user interface for the simulation), for each user, the social network selects one article (i.e., news source) that is presented to the user. This simulates the case where the social network may, for example, recommend to the user a link to one suggested article to consider reading. We then determine if the user reads this article or not, which is determined by the probability of the user reading an article from the news source that this article comes from. In the simulation, the social network keeps track of the number of articles from each news source that are presented to each user as well as the number of articles from each news source that are read by each user.

Keeping track of the number of articles presented to and read by each user helps the social network determine an estimate of the probability (also referred to as the "affinity") that a particular user will read an article presented from a particular news site. Recall that the social network does not know each user's true probabilities of reading articles from different news sources, so it must estimate this value from data in order to determine what the user is likely to read in the future. The social network models the probability that a user will read an article from a particular news source by simply computing the percentage of articles read by the user from that news source, given by the formula:

$$\frac{\text{Number of articles read by the user from news source}}{\text{Number of articles presented to the user from news source}}$$

Note that when the simulation starts, the social network assumes that every user has been presented with 2 articles from each news source and that 1 article from each news source was read[1]. So, before we get any actual data, the social network simply estimates that a user has a 50% chance of reading an article from any news source.

Selection of initial news article for each user (each day)

The selection of the article (i.e., news source) that the social network decides to present to a user each day can be made either by "exploring" the set of all news courses or "exploiting" the news source that the social network believes the user is mostly likely to read. The probability (a real value between 0 and 1) of choosing the "explore" option is set as a parameter called "Probability to explore for one user" in the user interface at the start of the simulation.

If the "explore" option is selected, the social network selects an article (news source) randomly from among all the news sources, where the chance of picking a particular news source is weighted by the likelihood that we believe the user will read an article from that news source. For example, say the social network currently has the following estimates that a particular user will read an article from the different news sources:

[1]The reader familiar with probability estimation might recognize this as a Laplace or Beta(1, 1) prior probability.

Source	1	2	3	4	5	6	7	8	9	10
P(read)	0.25	0.25	0.85	0.50	0.25	0.50	0.25	0.25	0.50	0.40

Then, we would select one of the news sources with the following (proportional) probabilities (rounded to two decimal places):

Source	1	2	3	4	5	6	7	8	9	10
P(read)	0.06	0.06	0.21	0.13	0.06	0.13	0.06	0.06	0.13	0.10

If the "exploit" option is selection (i.e., we did not select the "explore" option), then the social network simply selects the news source which has the highest estimated probability of being read by the user. For the example user given above, that would be news source 3.

Note that the decision to "explore" or "exploit" is made separately for each user on each day of the simulation.

Once an article is selected for a user, we then determine if the user reads that article or not (based on the user's true probability of reading an article from that news source). Each article that is thus read may then potentially appear in the news feed of other users in the social network as reading the article causes it to be posted to the network by the user.

Simulating users' news feeds in the social network

After determining which initial articles (news sources) are read by each user (and thus posted in the social network) on a particular day, we then determine which of these articles should appear in the news feeds of other users in the social network.

Each user in the network can see a maximum of 10 articles posted by friends in each day of the simulation. Recall that all users in the simulation are friends with each other. The choice of each of the articles that are shown to a particular user (referred to here as User X) can be made either by "exploring" or "exploiting" the affinity between User X and other users in the social network.

The affinity between User X and some other user (call them User Y) in the social network is simply the percentage of time that articles shared between User X and User Y are read by the other user. In this simulation, we don't distinguish if an article from User X is posted to User Y's news feed, or vice versa. We simply keep track of the total number of articles that either were posted to User Y's news feed from User X or were posted to User X's news feed from User Y. Likewise, we also track the number of such posted articles that were read by the user to whose news feed they were posted to. The percentage of read articles (out of all posted articles) is the affinity between User X and User Y (and, by symmetry, is the same as the affinity between User Y and User X).

The probability (a real value between 0 and 1) of choosing the "exploring" option (which is also a measure of the *diversity* among users from which news feed posts are selected) is set as a parameter called "Probability of diversity among users" in the user interface at the start of the simulation. For each of the (up to 10) posts that may appear in User X's news feed, we separately determine if the posting should be chosen via the "explore" or "exploit" option.

If the "explore" option is chosen for a posting, we select an article to include in the news feed for User X randomly from among all articles initially read that day by other users in the network (which have not previously been posted in User X's news feed). The random selection of this article is weighted by the affinity of User X with each other user in the network. In other words, the probability of picking an article to show from User Y to show in User X's news feed is proportional to the affinity of User X with User Y. For example, say User X has five friends in the network that read an article at the beginning of the day and the affinity of User X with these five other users is (respectively):

User	1	2	3	4	5
Affinity	0.6	0.4	0.7	0.2	0.1

Then, we would select a posting from among the five users with respective probabilities:

User	1	2	3	4	5
Affinity	0.3	0.2	0.35	0.1	0.05

If the "exploit" option is chosen for a posting, we select an article to include in the news feed for User

X that comes from whichever user that has the highest affinity with User X (and hasn't already been selected previously for a posting in User X's news feed). So, in the example shown above, we would select the article from User 3 to post in User X's news feed, since User 3 has the highest affinity with User X.

After determining the set of articles that are posted in the news feed for each user in the social network, we then determine which of these articles are read by each user, respectively. The chance of a user reading an article posted to their feed is determined by the true probability of the user reading an article from the news source of the article. We keep track of the number of articles that were posted to a user's news feed from all other users in the network as well as which of the articles were read in order to update the affinity values between every pair of users in the social network.

Running the simulation

The file `networksimulation.py` is the main program file that runs the simulation. The program starts by asking the user for various parameters, such as the number of left and right-leaning users in the simulation and the probability "to explore for one user" as well as the probability "of diversity among users." After the user provides those parameters, the simulation runs and prints out information about the evolution of the social network over time (explained in detail below).

The results of the simulation

As the simulation runs, it prints updates about the links (affinities) between users of different types (left-leaning and right-leaning) in the network over time. More specifically, every `DAYS_PER_UPDATE` days in the simulation, the simulation reports the number of links that are strong (affinity ¿ 0.6), medium (affinity 0.45-0.6), weak (affinity 0.4-0.45), or very weak (affinity ¡ 0.4) between pairs of users, where each pair may be composed of two left-leaning users, two right-leaning users, or one left-leaning and one right-leaning user.

For example, below we show that at the start (day 0) of a simulation (whose parameters are also given below), there are 105 medium links between pairs of left-leaning users, 105 medium links between pairs of right-leaning users, and 225 medium links between pairs of one left-leaning and one-right leaning user. Note that with 15 left-leaning users, there are 105 (= 15 * 14 / 2) possible distinct pairs of left-leaning users. Since there are also 15 right-leaning users, there is the same number of pairs of two right-leaning users. Also, there are 225 (= 15 * 15) possible distinct pairs composed of one left-leaning and one right-leaning user. There are only medium strength links in the network at this point as all links between pairs of users start with 0.5 affinity.

```
Number of left-leaning users: 15
Number of right-leaning users: 15
Number of days in the simulation: 500
Probability to explore for one user: 0.7
Probability of diversity among users: 0.3

Day #0:
Strong links:
 Number of links between two left-leaning users: 0
 Number of links between two right-leaning users: 0
 Number of links between one left-leaning user and one right-leaning user: 0
Medium links:
 Number of links between two left-leaning users: 105
 Number of links between two right-leaning users: 105
 Number of links between one left-leaning user and one right-leaning user: 225
Weak links:
 Number of links between two left-leaning users: 0
 Number of links between two right-leaning users: 0
 Number of links between one left-leaning user and one right-leaning user: 0
Very Weak links:
 Number of links between two left-leaning users: 0
 Number of links between two right-leaning users: 0
 Number of links between one left-leaning user and one right-leaning user: 0
```

By the last day (day 500) in the simulation, the structure of the network has changed, as shown below:

```
Day #500:
Strong links:
 Number of links between two left-leaning users: 66
 Number of links between two right-leaning users: 62
 Number of links between one left-leaning user and one right-leaning user: 0
Medium links:
 Number of links between two left-leaning users: 39
 Number of links between two right-leaning users: 43
 Number of links between one left-leaning user and one right-leaning user: 4
Weak links:
 Number of links between two left-leaning users: 0
 Number of links between two right-leaning users: 0
 Number of links between one left-leaning user and one right-leaning user: 75
Very Weak links:
 Number of links between two left-leaning users: 0
 Number of links between two right-leaning users: 0
 Number of links between one left-leaning user and one right-leaning user: 146

Total articles shown: 150791
Total articles read: 82828
Percentage read: 54.93%
Total revenue: $4141.40
```

At this point, we see that there are 66 pairs of two left-leaning users that have strong links between them as well as 62 pairs of two right-leaning users with strong links. And, many of the pairs composed of one left-leaning and one-right leaning users now only have weak or very weak link strength.

The simulation also reports the total number of articles shown during the simulation (to all users), the total number of articles read during the simulation (by all users), the percentage of the articles shown that were read, and the total revenue generated by all the read articles. Note that the total revenue generated is simply the number of articles read multiplied by $0.05, which is a stylized estimate of the monetization value for a user interacting with a piece of content on the web (e.g., payment for clicking on an ad or referral fee for having a user click on an article). The text printed as a result of the simulation can be copied/pasted for use in your write-up.

Acknowledgments This assignment was written and designed by Rob Reich, Jeremy Weinstein, and Mehran Sahami, all at the Stanford University for their CS182 course. Student Michael Dworsky also categorized the recidivism data and developed an early prototype for the assignment. They have given me permission to use their materials.
Any changes made have been minor with the aim of providing context, or adapting to the specifics of CSC-395 at Grinnell College.