Ethics, Public Policy, and Technological Change

Rob Reich
Mehran Sahami
Head TA: Roberta Fischli

Housekeeping

- Assignment #3 is out today on the CS182 website and is due 11:59pm on Thursday, February 26th
- To get you prepared, we'll discuss several aspects of the assignment today

Today's Agenda

1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #3: Social network simulation
5. Perspectives on filter bubbles

Today's Agenda

1. **Formal representations for networks**
2. Network structures
3. Models for content recommendation
4. Assignment #3: Social network simulation
5. Perspectives on filter bubbles

The Power of Network Effects

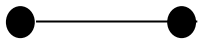
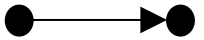
- The power of platforms often comes from network effects
  - The more entities in a network, the more useful it is for everyone in it
  - Classic example: telephone network
  - Modern examples: websites on the Internet, people in a social network, buyers/sellers in a marketplace
- Platform network effects also create monopolistic effects
  - What is the #1 online auction website?
  - What is #2?
  - Sellers want to sell at the marketplace with the most buyers
  - Buyers want to buy at the marketplace with the most sellers
- Formally representing networks allows us to better understand how they work and analyze their dynamics

Networks as Graphs

- Formally: Graph G = (V, E), where
  - V: a non-empty set of “vertices” (or “nodes”)
  - E: set of “edges” defined by pairs of distinct elements of V
- Examples:

- On the web
  - Vertices are webpages; edges are links between webpages
- In a social network
  - Vertices are users; edges represent friendships between users

Graph Terminology

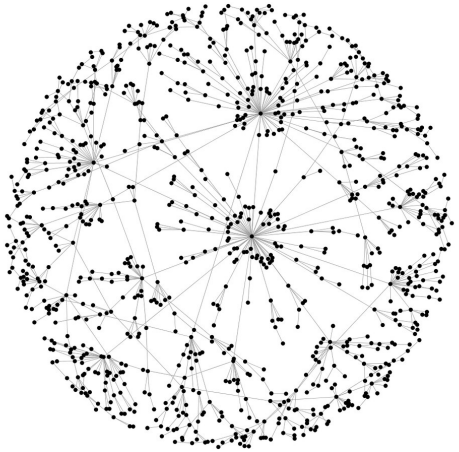
- Undirected (simple) graphs:
  - Pairs of vertices in E representing edges are unordered 
  - Reciprocal relationship (e.g., co-authors of paper, friends in social network)
  - "Degree of vertex": number of edges connected to it
- Directed graphs:
  - Pairs of vertices in E representing edges are ordered (start, end) and called "arcs" 
  - Relationship is unidirectional (e.g., webpage links to another webpage)
  - "In-degree of vertex": number of edges pointing into it
  - "Out-degree of vertex": number of edges pointing out of it
- Weighted graphs:
  - Elements in E (edges) are weighted by a real value
  - E.g., Affinity between two users in a social network

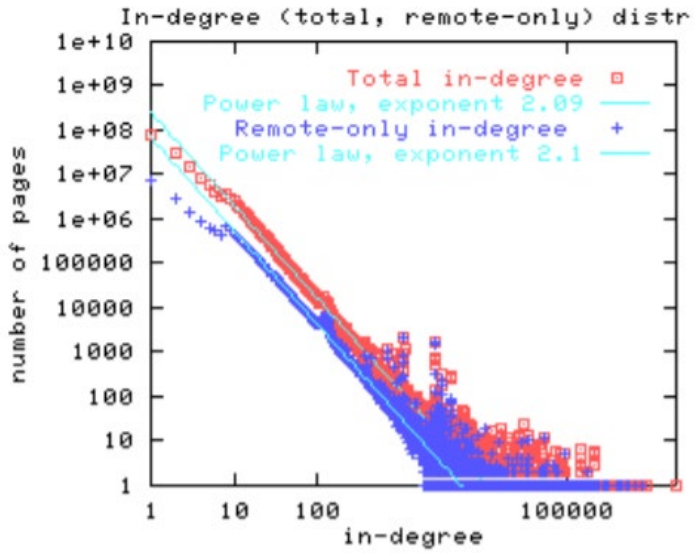
Today's Agenda

1. Formal representations for networks
2. **Network structures**
3. Models for content recommendation
4. Assignment #3: Social network simulation
5. Perspectives on filter bubbles

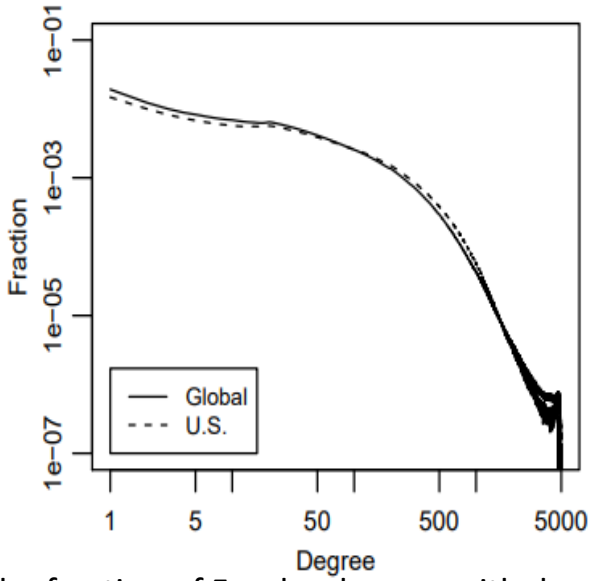
Scale-Free Networks

- Scale-Free Networks
  - Contain few vertices with high degree
  - Contain many vertices with low degree
  - Examples:
    - Connectivity of pages on the web
    - Friend relationships in Facebook





Distribution of web pages by in-degree (Broder *et al*, 2000)



The fraction of Facebook users with degree *k* (Ugander *et al*, 2011)

Top right image: Flickr, CC BY 2.0

How Do Scale-Free Networks Arise?

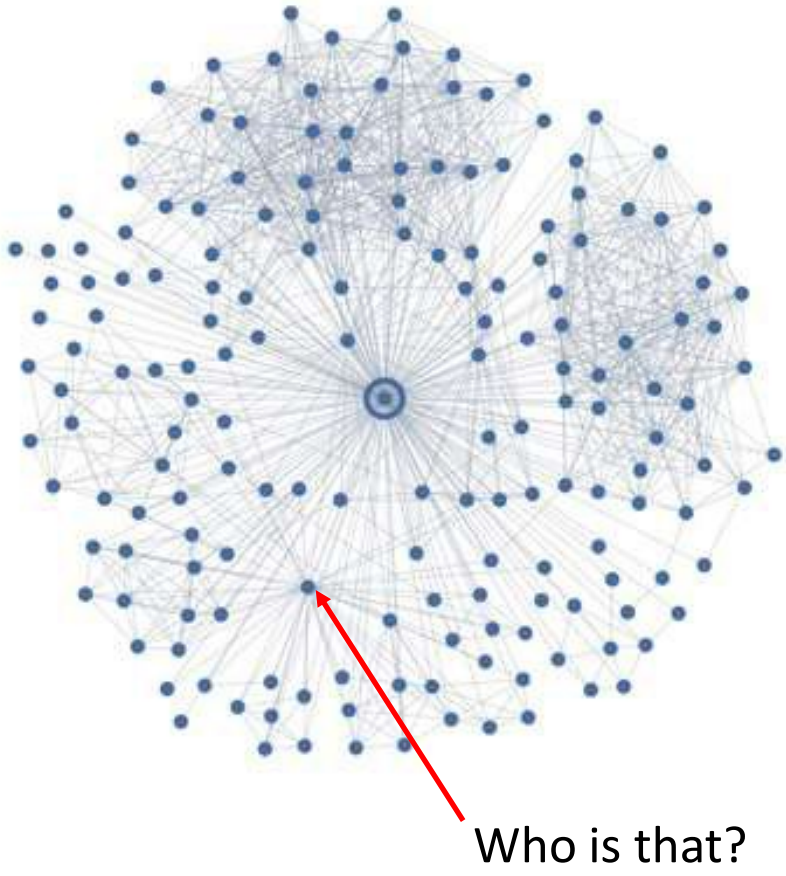
- Preferential attachment (Barabási and Albert, 1999)
  - When new vertices are added to a network, their probability of linking to another vertex depends on degree of that vertex
  - "Rich get richer"
- On the web
  - If a web page has lots of links, it is more likely to get linked to
  - If you are a well-known web page (as a result of having many links), others are more likely to be able to find and link to you
- In a social network
  - If you have a lot of friends on Facebook/Instagram/etc, you are likely to get even more friends in the future
  - When you have many friends, easier for others to find you or get recommendation to add you as a friend
  - There are caveats (e.g., maximum number of friends)

What Graph Structures Can Reveal

- Analyzing romantic partnerships (Backstrom and Kleinberg, 2013)
  - Consider friendship graph for user (vertex circled at center)

- Correctly identified a user's romantic partner (e.g., spouse, fiancé, partner) with 70.5% accuracy based on analyzing network structure

- "We find that relationships on which recursive dispersion fails to correctly identify the partner are significantly more likely to transition to 'single' status over a 60-day period."



Today's Agenda

1. Formal representations for networks
2. Network structures
3. **Models for content recommendation**
4. Assignment #3: Social network simulation
5. Perspectives on filter bubbles

Models for Recommendation

- Given information about individual users on a platform, want to make content recommendations to those users
- Netflix
  - Recommend movies that you might enjoy watching
- Facebook/Instagram
  - Recommend people you might want to be friends with
  - Show content in your feed based on what friends have posted/read
- Google
  - Show ads based on current (and prior) search history
- Many ways to approach this problem
  - We'll consider two here

Clustering-Based Approaches

- Model each user as an instance vector of values
  - E.g., ratings given to movies watched in the past
- Cluster user vectors to find groups of “like-minded” users
  - Requires a way to measure similarity between user instances
    - Measure is usually customized for the application
- Determine items (e.g. movies, products, etc.) that other users in the same cluster as you rated highly (in aggregate)
  - Recommend items to you that others in your cluster ranked highly
  - Called “collaborative filtering”
- Open questions: How many clusters to form? What similarity measure to use? Specificity vs. generality (e.g. recommend Star Wars to everyone)?

Direct Affinity Measurement

- Model direct relationships between users/entities
  - How often two users click on ("like") each others posts
  - How often a user reads an article suggested from a particular site
  - How often a video is viewed after another video (or set of videos)
- Measure interactions between users/entities in the system
  - Simple methods: compute percentage of time some interaction occurs
  - Complex methods: use machine learning to predict likelihood of some interaction occurring based on a set of observations
- Make a recommendation based on the (highest) likelihood of the user making an interaction based on the recommendation
  - E.g., Recommend a post from a friend whose posts you often “like”
  - Which interactions to model? How to make explore/exploit trade-off?
- Choices of data to capture, inferences to make, and models to use matter! (e.g., Google vs. Yahoo)

Exploration/Exploitation Trade-off

- In recommender systems, we need to gather data about user preferences in order to make more accurate recommendations
  - When making a recommendation we can choose to either "explore" or "exploit" with the recommendation
  - E.g., How do I make good movie recommendations if you've seen/rated few (or even no) movies? This is called the "cold start" problem.
- "Explore"
  - Make recommendation that may **not** have highest probability for user engagement (i.e., choose randomly, using some weighting)
  - Intent: gather more data to help make future predictions more accurate
- "Exploit"
  - Make recommendation with highest probability for user engagement
  - Intent: have the user follow the recommendation

Modeling User Affinity

- Consider we want to model how likely (e.g., percentage of the time) user X reads a posting in their feed from friend Y
  - We keep track of the number of times we posted an item from Y in the news feed of friend X (call this the number of items "presented")
  - Also keep track of the number of times user X interacted with (e.g., clicked on, "liked", read, etc.) a posting from user Y (call this the number of items "read")
  - A simple measure of the affinity between user X and Y is fraction: (Number of items read / Number of items presented)
- "Cold start" problem
  - When we start, we have no data on interaction of X and Y
  - Simple solution—initialize: Number of items presented = 2 and Number of items read = 1 (this is also known as a "Laplace prior")
    - Without real data, estimate that X has 50% chance of reading Y's post
    - Update "presented" and "read" numbers as we get actual data

Updating User Affinity

- Consider affinity between two entities X and Y
  - Here, we will treat affinity between X and Y symmetrically.
    - That is, it doesn't matter who is producer and who is consumer
    - Sometimes, might model X → Y and Y → X separately
  - Initialize "Number presented" and "Number read"
  - Item (article) from X presented to Y; article not read by Y
  - Item (article) from Y presented to X; article read by X
  - Item (article) from Y presented to X; article read by X
  - Item (article) from X presented to Y; article read by Y
  - Item (article) from Y presented to X; article not read by X

Number presented	Number read	Affinity
7	4	0.57

Today's Agenda

1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. **Assignment #3: Social network simulation**
5. Perspectives on filter bubbles

Where Do You Get Your News?

- Where do you get your news?
  - Subscriptions to specific papers/magazines?
  - Social networks (e.g., Facebook, Instagram, Twitter/X)?
  - Video (e.g., YouTube, TikTok)?
  - News aggregators (e.g., Google News, Yahoo News)?

- To what extent do you consume a diet of diverse news sources?

Social Network Simulation

- Want to examine the effect that content recommendation in a social network has with respect to political polarization
  - Polarization impacts free flow of information, causes “echo chambers” or “filter bubbles”, makes electorate less informed, etc.
- Simulate a set of users in social network
  - There is some number of “left-leaning” users and some number of “right-leaning” users in the network
  - All users in the network are friends (it’s a small network)
  - So, articles read by a user may be recommended to any other user
- There are 10 news sources in the simulation that the users may read articles from
  - News sources are considered on a spectrum from “left” to “right”
  - We use the term “article” to refer to a news source (i.e., the only aspect about an article that matters is which news source it comes from)

AllSides Media Bias Chart™

Ratings based on online, U.S. political content only – not TV, print, or radio.
Ratings do not reflect accuracy or credibility; they reflect perspective only.

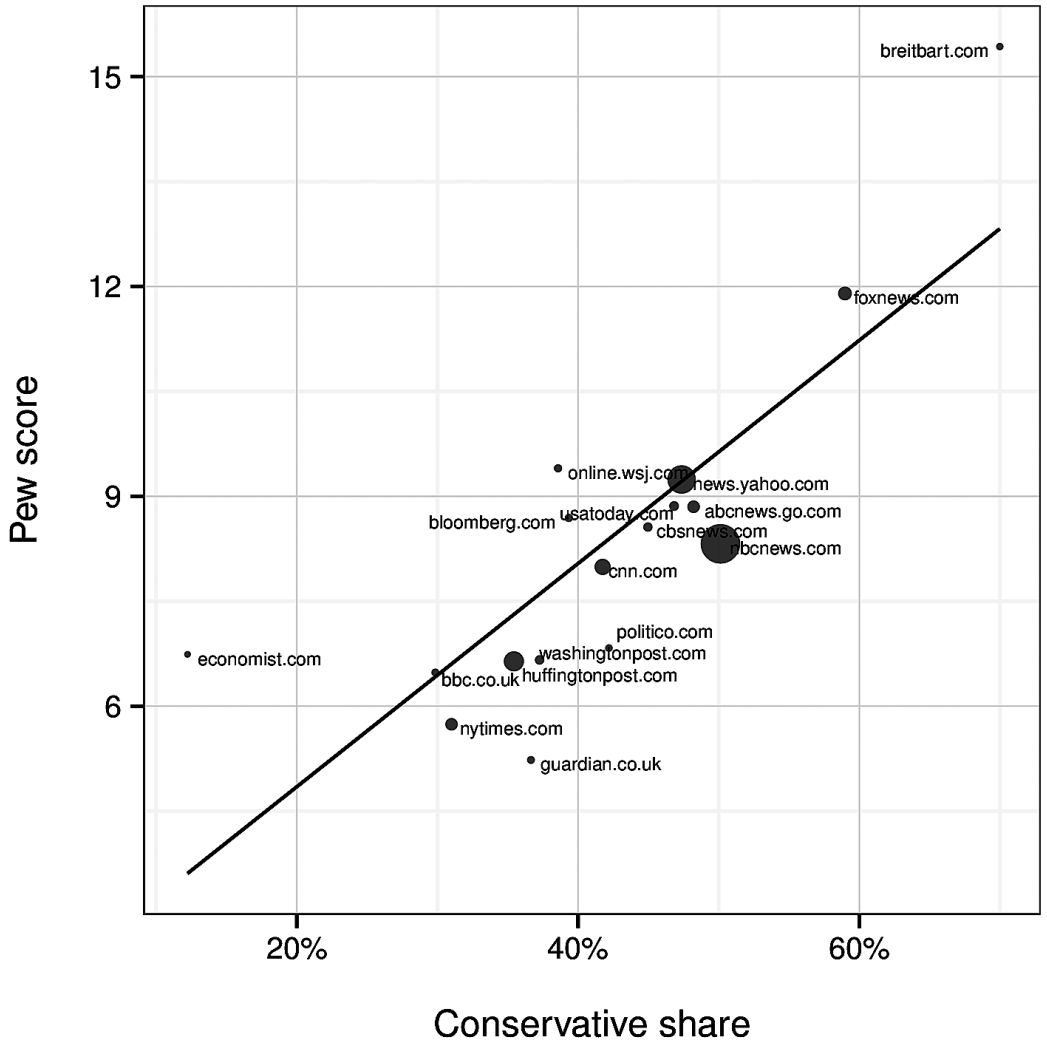
AllSides Media Bias Ratings™ are based on multi-partisan, scientific analysis.

Visit AllSides.com for balanced news and over 2,400 rated sources.

AllSides does not own the rights to third party logos.

Version 11

© AllSides 2025



[Flaxman, Goel, and Rao, 2016] A comparison of our estimate of conservative share of an outlet's audience to a Pew survey-based measure of audience ideology, where point sizes are proportional to popularity.

Model of Users Reading News

- Users each have some true probability for reading an article from a given news source.
  - True probabilities for reading news source for all "left-leaning" users are:

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

  - True probabilities for reading news source for all "right-leaning" users are:

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

- Each day in the simulation, for each user, we select one article (news source) that is presented to the user
  - Depending on the user's probability of reading an article from that news source, the user may then read the article or not.
  - We keep track of both how many articles from each news source are presented and read by a user to model the user's affinity for each source.

Model of Users Reading News

- We model affinity of each user for each news source using the percentage of articles presented from source that are read
- The selection of the initial article (each day) for each user can be made by either "exploring" or "exploiting"
  - If we "explore", we select a news source randomly, weighted by what have measured so far about the user's likelihood of reading a source.
  - Say, our current estimate of a user reading news sources is:

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 a news source with the following probabilities:

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

  - If we "exploit", we select news source with highest probability of being read based on data we have (which is source 3 for the user above)
  - You set probability of "exploring" (*Probability to explore for one user*)

Model of Users Reading News

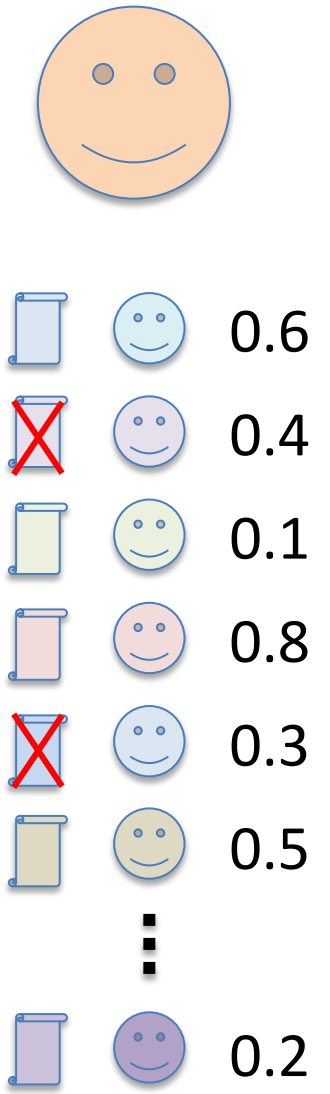
- In other words… Each day, for each user
  - We flip a coin
  - Probability of "heads" is set as *Probability to explore for one user*
  - If we get "heads" (explore), we select a news source randomly for user
    - Probability for each news source being selected is based on what we've measured so far of that user's likelihood of reading source
  - If we get "tails" (exploit), we select news source with that has highest probability of being read by the user
    - Again, probability for each news source is based on what we've measured as that user's likelihood of reading each news source
  - No matter how the article is chosen, we then flip another coin to determine if the user reads the article or not
    - Probability for reading article determined by user's actual probability to read that news source (and we keep track of this)

Model of Users' News Feeds

- After determining which initial articles (for each user) were read, we then determine which of these read articles should appear in the news feed for other users in the network
- Each user (each day) can be presented a maximum of 10 articles in their news feed
  - The choice of each article to show in user X's news feed can be made by either "exploring" or "exploiting"
  - If we "explore", we select an article to include in news feed of user X by choosing randomly from among articles read that day by other users in the network, weighted by what we have measured so far about user X's affinity for reading articles from other users
  - If we "exploit", we select the article (news source) read by user Y who has highest affinity with user X for reading articles
  - You set probability of "exploring" (*Probability of diversity among users*)

Model of Users' News Feeds

- Each user (each day) can be presented a maximum of 10 articles in their news feed
  - For each of the 10 articles for user X, we flip a coin
  - Probability of "heads" is set as *Probability of diversity among users*
  - If we get "heads" (explore), we select article <u>randomly</u> for user X from articles their friends read that day
    - Probability of selecting article is determined by weighting of user X's affinity with each other user
  - If we get "tails" (exploit), we select article read by user Y, who has <u>highest</u> affinity with user X
  - Articles in the feed are not duplicated
    - So, if article from highest affinity user is already in feed, we'll take article from next highest affinity user



Model of Users' News Feeds

- After determining which articles appear in the news feed for each user, we then determine which of those articles the user reads (based on the user's probability of reading the source for each article)
  - We keep track of the number of articles originally read by user X that were presented in user Y's news feed, and whether the articles were read by user Y
  - This allow us to update the affinity between user X and Y over time
- The affinity between users X and Y is symmetric
  - That is, we don't care who was the poster and who was the reader
  - We only care if an article read by one of those two users was read by the other user (after appearing in their news feed)

Displaying the Results

- Network simulation program measure affinity between users in social network
  - Edges are labelled by strength of affinity between pair of users
    - Strong link (> 0.6)
    - Medium link (0.45-0.6)
    - Weak link (0.4-0.45)
    - Very Weak link (< 0.4)

- Additional information:
  - Total articles shown (to all users)
  - Total articles read (to all users)
  - Percentage of articles read
  - Total revenue: $0.05 **x** (# read)

Demo

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

Number of right-leaning users: 15
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

Displaying the Results

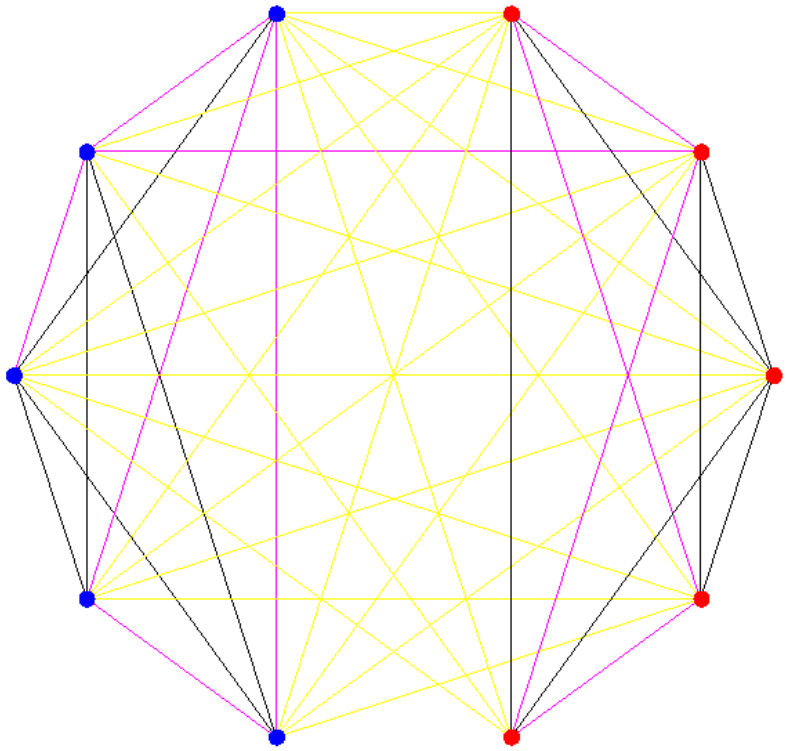
Number of left-leaning users: 15
Probability to explore for one user: 0.7
Number of days in the simulation: 500

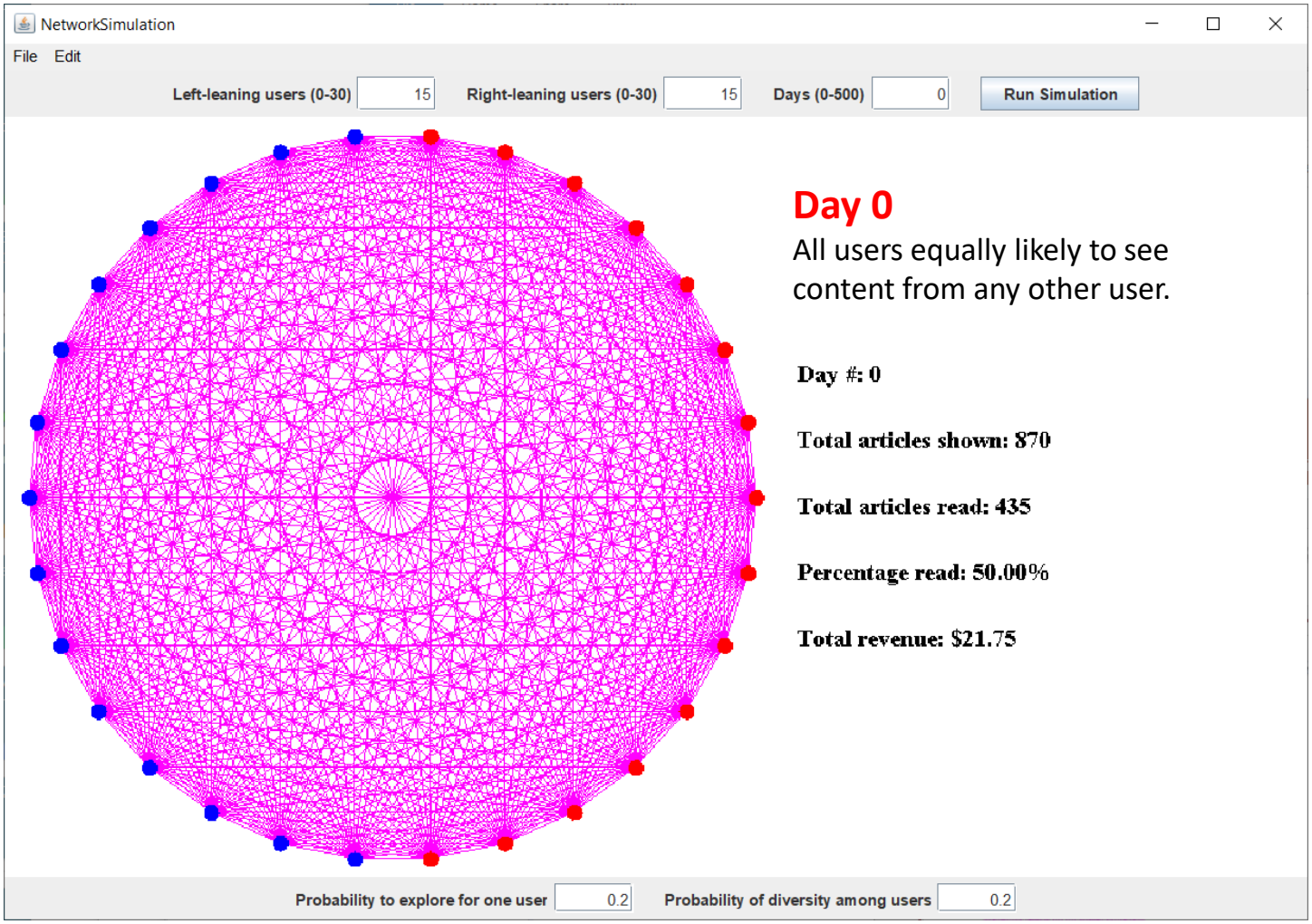
Number of right-leaning users: 15
Probability of diversity among users: 0.3

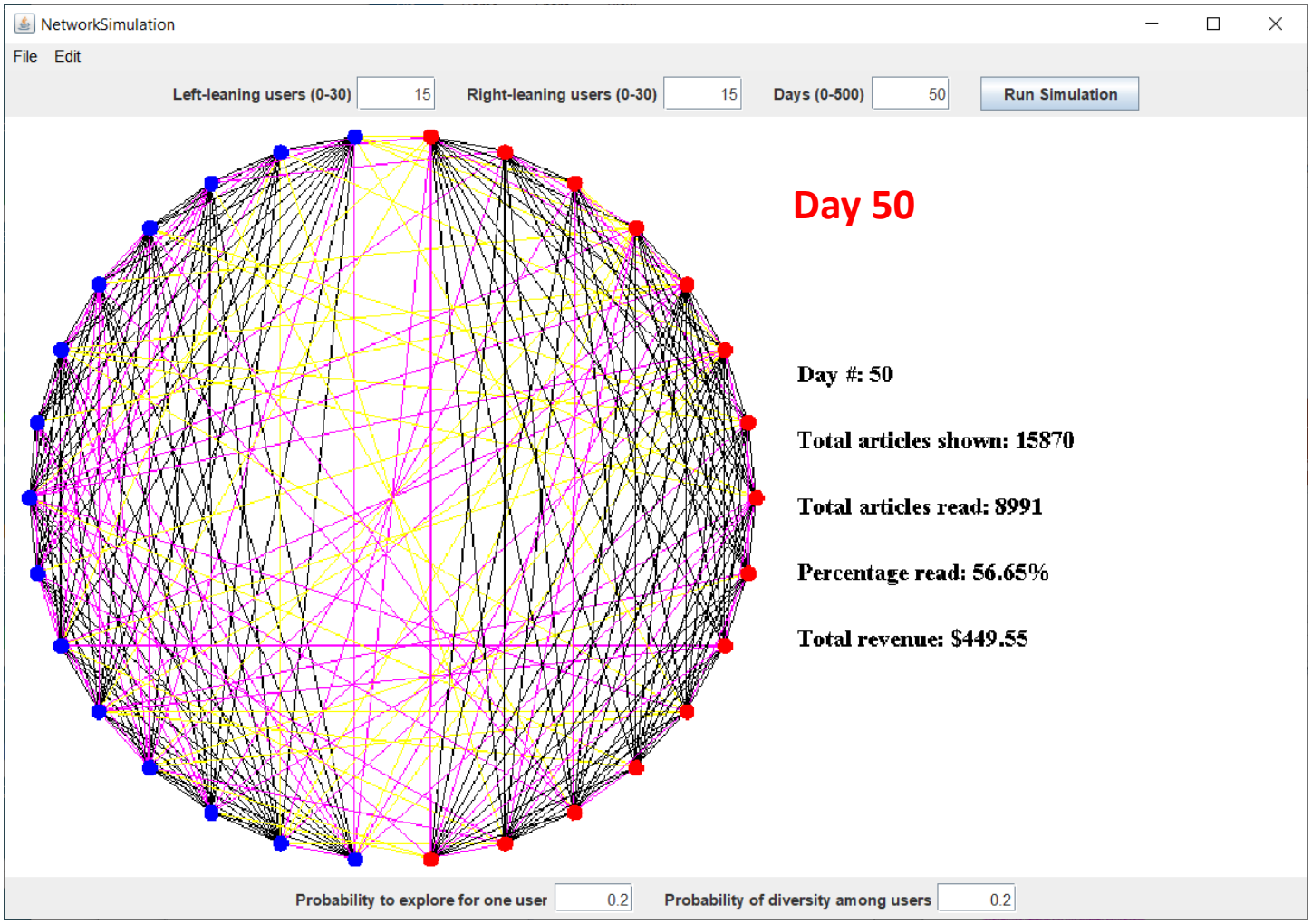
Day #:500
Strong links:
 Number of links between two left-leaning users: 61
 Number of links between two right-leaning users: 79
 Number of links between one left-leaning user and one right-leaning user: 0
Medium links:
 Number of links between two left-leaning users: 44
 Number of links between two right-leaning users: 26
 Number of links between one left-leaning user and one right-leaning user: 5
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: 96
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: 124

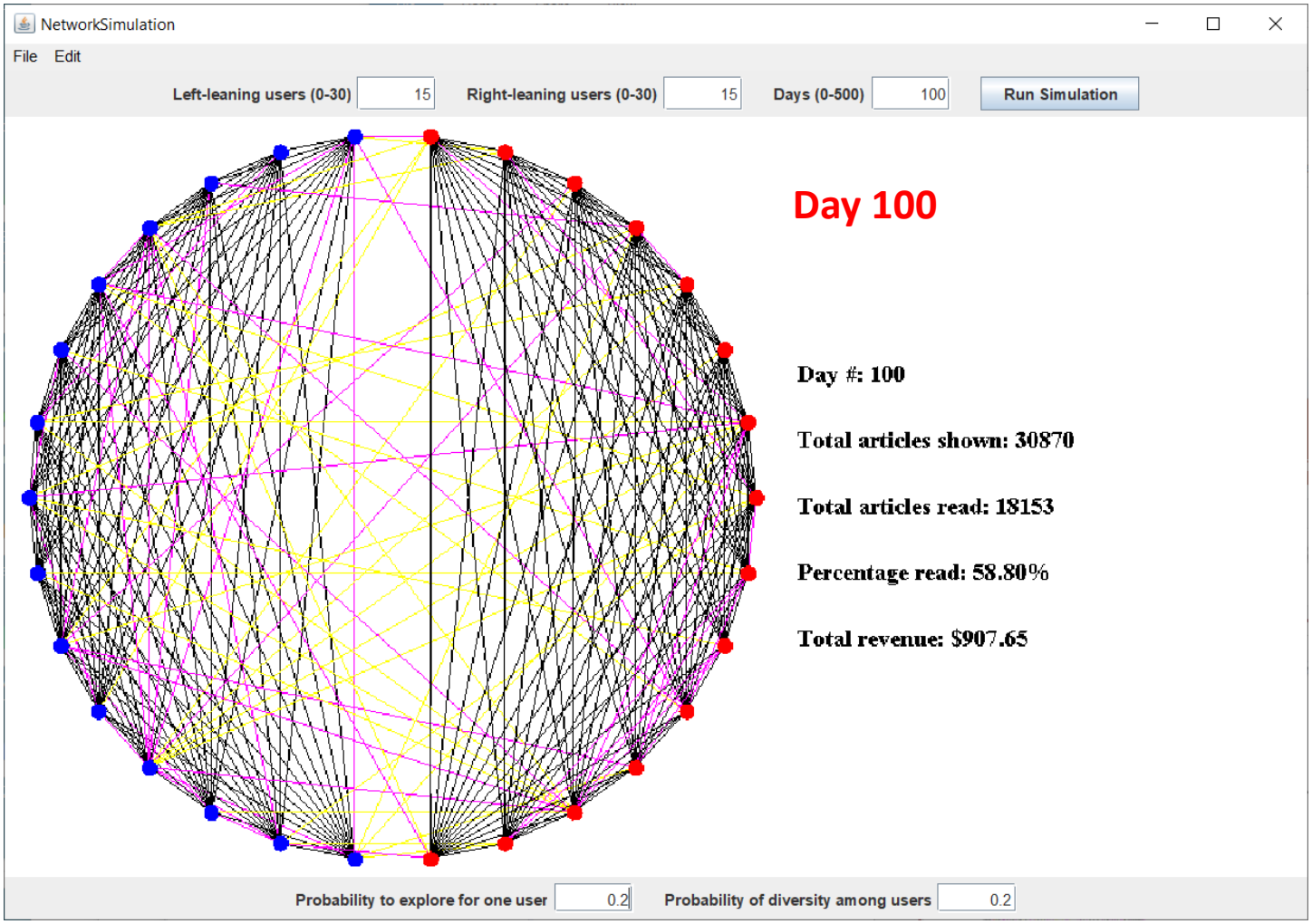
Displaying the Results

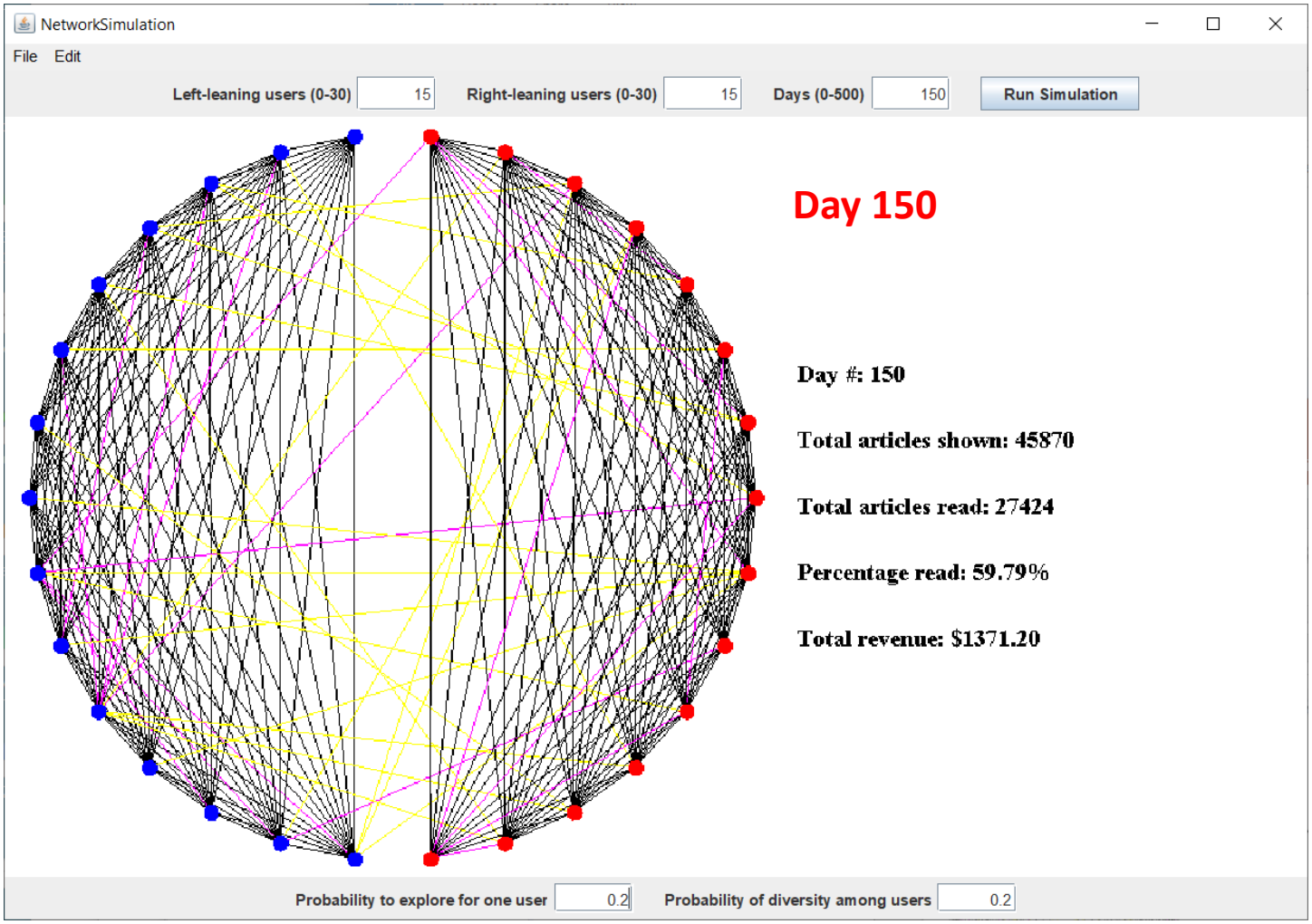
- To show network evolution, display social network as a graph
  - Users are vertices (circles), color-coded by political leaning
  - Edges are color-coded by strength of affinity between pair of users
    - *Black* = strong (> 0.6)
    - *Magenta* = medium (0.45-0.6)
    - *Yellow* = low (0.4-0.45)
    - no edge = very low (< 0.4)
  - Graph is just for in-class demo
  - Simulation in assignment will just give text results (shown earlier)
    - Easier to cut/paste or refer to in write-up

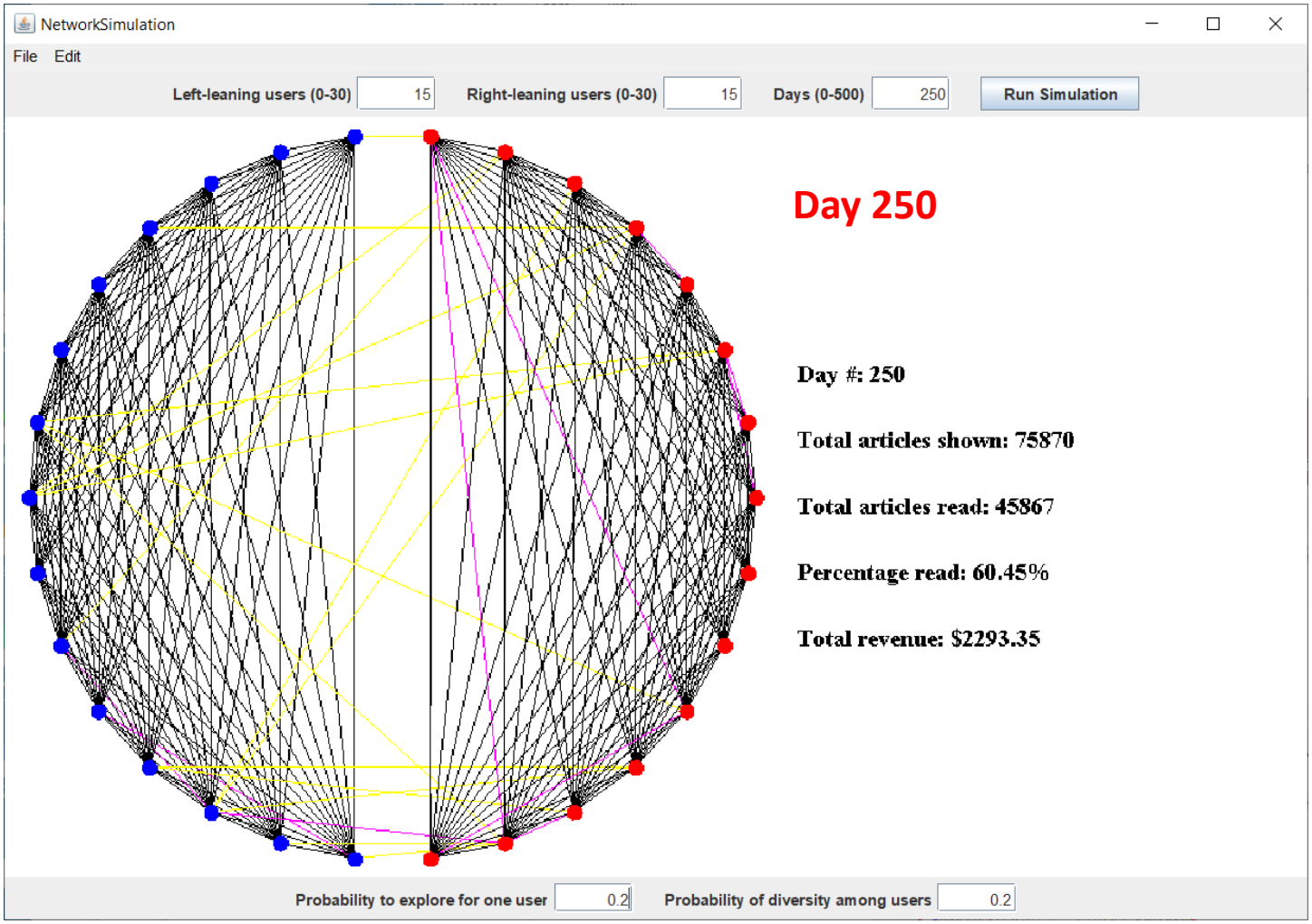


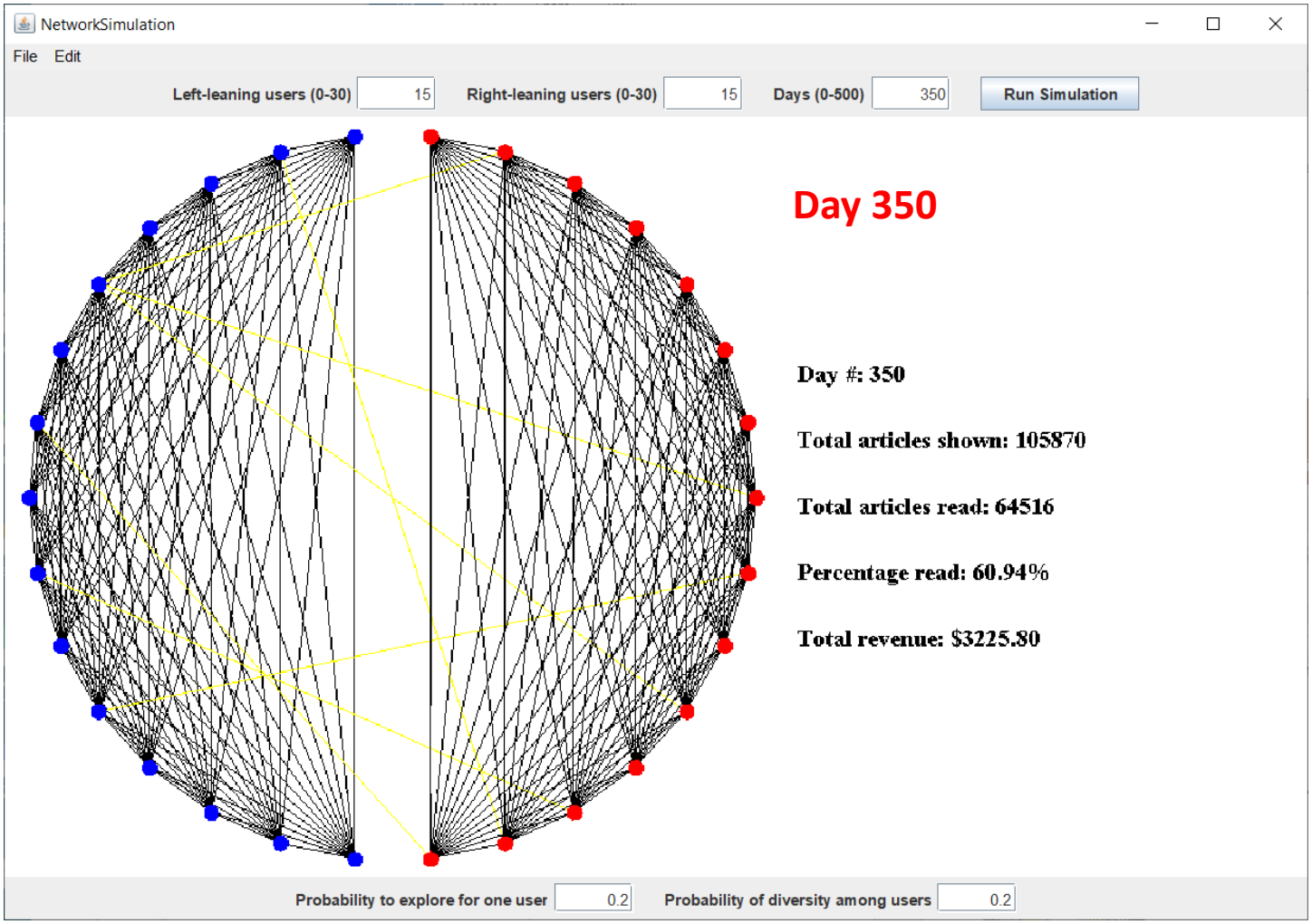

NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
0
Run Simulation
Day 0
All users equally likely to see content from any other user.
Day #: 0
Total articles shown: 870
Total articles read: 435
Percentage read: 50.00%
Total revenue: $21.75
Probability to explore for one user
0.2
Probability of diversity among users
0.2

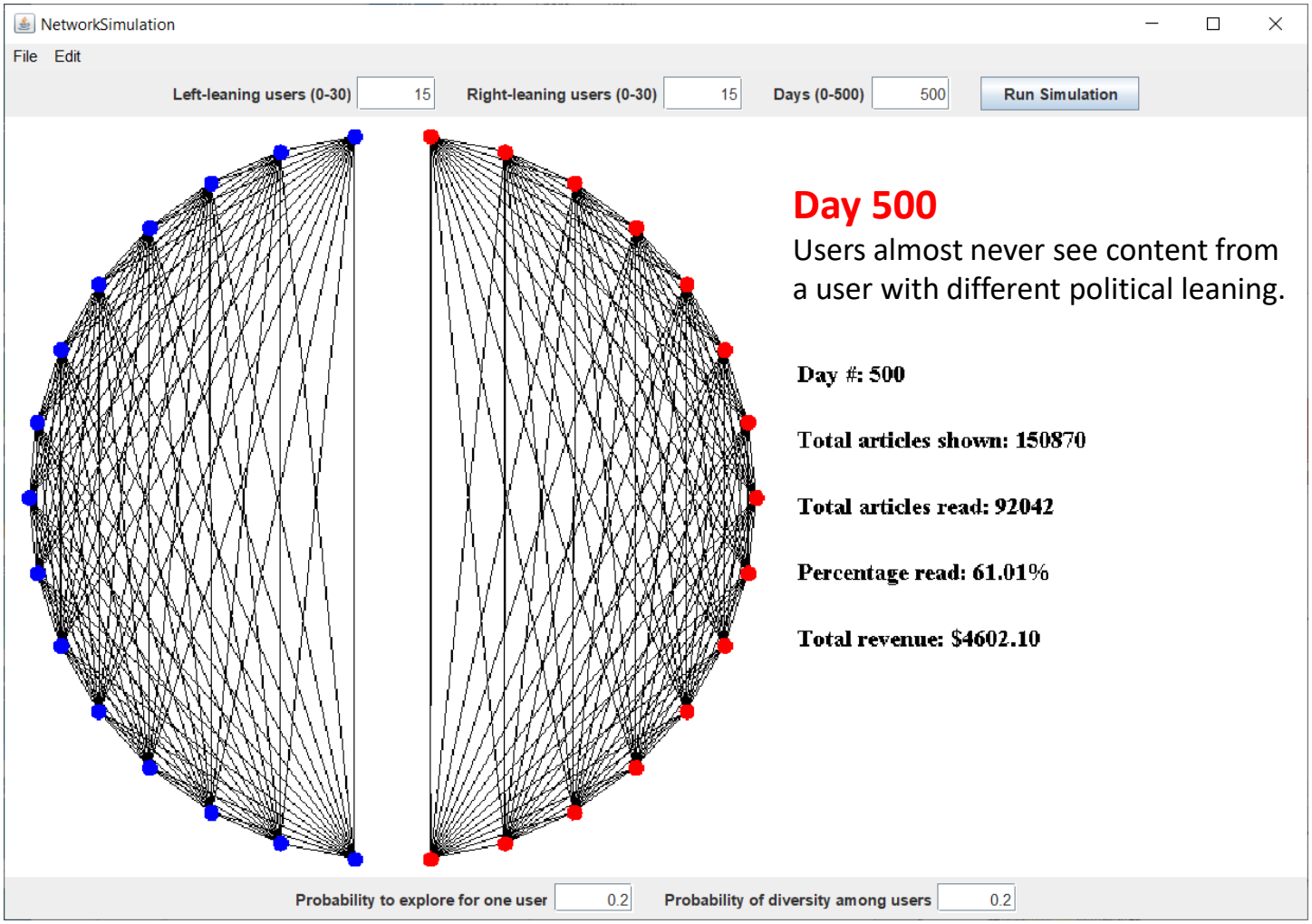

NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
50
Run Simulation
Day 50
Day #: 50
Total articles shown: 15870
Total articles read: 8991
Percentage read: 56.65%
Total revenue: $449.55
Probability to explore for one user
0.2
Probability of diversity among users
0.2


NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
100
Run Simulation
Day 100
Day #: 100
Total articles shown: 30870
Total articles read: 18153
Percentage read: 58.80%
Total revenue: $907.65
Probability to explore for one user
0.2
Probability of diversity among users
0.2


NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
150
Run Simulation
Day 150
Day #: 150
Total articles shown: 45870
Total articles read: 27424
Percentage read: 59.79%
Total revenue: $1371.20
Probability to explore for one user
0.2
Probability of diversity among users
0.2


NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
250
Run Simulation
Day 250
Day #: 250
Total articles shown: 75870
Total articles read: 45867
Percentage read: 60.45%
Total revenue: $2293.35
Probability to explore for one user
0.2
Probability of diversity among users
0.2


NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
350
Run Simulation
Day 350
Day #: 350
Total articles shown: 105870
Total articles read: 64516
Percentage read: 60.94%
Total revenue: $3225.80
Probability to explore for one user
0.2
Probability of diversity among users
0.2

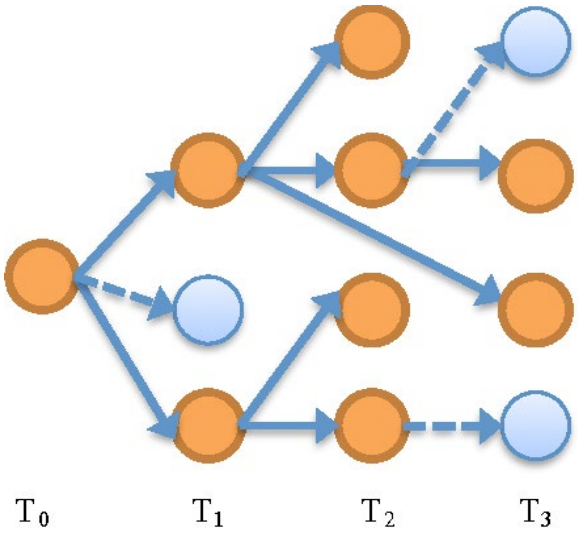

NetworkSimulation
File
Edit
Left-leaning users (0-30)
15
Right-leaning users (0-30)
15
Days (0-500)
500
Run Simulation
Day 500
Users almost never see content from a user with different political leaning.
Day #: 500
Total articles shown: 150870
Total articles read: 92042
Percentage read: 61.01%
Total revenue: $4602.10
Probability to explore for one user
0.2
Probability of diversity among users
0.2

Today's Agenda

1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #3: Social network simulation
5. **Perspectives on filter bubbles**

Perspectives on Filter Bubbles

- In our simulation, we don't consider "reposting" for simplicity
  - Each article read at the beginning of the day by user can only appear in other users news feeds that day and cannot be reposted by others
  - In many real contexts (e.g., Facebook, Twitter), information flows in "cascades" over time
  - Would further exacerbate polarization in our simulation (and make it harder to understand)
- Cascades often modeled are "contagion"
  - A posting has a probability of "infecting" reader and being reposted
  - Phenomenon in exploited for "viral marketing"
    - E.g., "Influencers"
    - *Story time:* Hotmail



Source: Hakim and Khodra, 2014

Image: "Predicting Image Cascade on Twitter," Hakim and Khodra, 2014. https://www.semanticscholar.org/paper/Predicting-information-cascade-on-Twitter-using-Hakim-Khodra/4d0dfb11a6e81559602cf32affe9da118a40adc3/figure/0