CS 182: Ethics, Public Policy, and Technological Change

Rob Reich
Mehran Sahami
Jeremy Weinstein
Ece Korkmaz (Co-Head TA)
Alena Smith (Co-Head TA)
Assignment #4 out today on the CS182 website and is due 11:59pm on Thursday, March 9th

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 #4: 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 #4: Social network simulation
5. Perspectives on filter bubbles
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
1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #4: 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)
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, 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 #4: Social network simulation
5. Perspectives on filter bubbles
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
  - 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
  - Recall clustering as a form of “unsupervised learning” covered in AI unit

- 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 other’s 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)
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?

“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
• 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 $X \rightarrow Y$ and $Y \rightarrow X$ 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

<table>
<thead>
<tr>
<th>Number presented</th>
<th>Number read</th>
<th>Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>0.57</td>
</tr>
</tbody>
</table>
Today’s Agenda

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

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(read)</td>
<td>0.70</td>
<td>0.66</td>
<td>0.61</td>
<td>0.56</td>
<td>0.52</td>
<td>0.48</td>
<td>0.43</td>
<td>0.39</td>
<td>0.34</td>
<td>0.30</td>
</tr>
</tbody>
</table>

• True probabilities for reading news source for all “right-leaning” users are:

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(read)</td>
<td>0.30</td>
<td>0.34</td>
<td>0.39</td>
<td>0.43</td>
<td>0.48</td>
<td>0.52</td>
<td>0.56</td>
<td>0.61</td>
<td>0.66</td>
<td>0.70</td>
</tr>
</tbody>
</table>

• 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.
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:

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(read)</td>
<td>0.25</td>
<td>0.25</td>
<td>0.85</td>
<td>0.50</td>
<td>0.25</td>
<td>0.50</td>
<td>0.25</td>
<td>0.25</td>
<td>0.50</td>
<td>0.40</td>
</tr>
</tbody>
</table>

- Then we would select a news source with the following probabilities:

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(select)</td>
<td>0.06</td>
<td>0.06</td>
<td>0.21</td>
<td>0.13</td>
<td>0.06</td>
<td>0.13</td>
<td>0.06</td>
<td>0.06</td>
<td>0.13</td>
<td>0.10</td>
</tr>
</tbody>
</table>

- 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’ 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).
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 allows 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 \times (# read)
Displaying the Results

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

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
Day #: 500
Total articles shown: 150870
Total articles read: 84147
Percentage read: 55.77%
Total revenue: $4207.35
Today’s Agenda

1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #4: 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

Echo Chambers in Real World

- Bakshy, Messing, and Adamic, 2015:
  - Analyzed data set of 10.1 million active U.S. Facebook users
  - Users self-reported ideological (political) affiliation

- Random: shared by random other person
- Potential from network: shared by friends
- Exposed: actually appeared in NewsFeed
- Selected: user clicked on

Broader Perspectives

• Information flow on platforms can be highly influenced by both explicit (diversity, banning) and implicit (optimization) decisions

• Should we be required to receive diverse information?
  • We used to: Federal Communications Commission Fairness Doctrine
    • Required licensed broadcasters to provide fair and balanced coverage of controversial issues
    • Introduced in 1949, abolished in 1987
  • Is there an obligation for platforms to show diverse information?
  • If so, how much and how should that diversity be determined?
    • E.g., Should a Google search for “New York Times” include pointers to other news outlets?
  • Is there an obligation for platforms to filter hate speech or “fake news”?
  • If so, how is determination of hate speech or “fakeness” made?
Comm. Decency Act, Section 230

“No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider.”

“No provider or user of an interactive computer service shall be held liable on account of—

(A) any action voluntarily taken in good faith to restrict access to or availability of material that the provider or user considers to be obscene, lewd, lascivious, filthy, excessively violent, harassing, or otherwise objectionable, whether or not such material is constitutionally protected”

• Electronic Frontier Foundation: “The Most Important Law Protecting Internet Free Speech”
Amending CDA 230: H.R. 2154

“Protecting Americans from Dangerous Algorithms Act”
- Introduced in House on March 23, 2021
- Co-sponsored by Anna Eshoo in CA, 18th District (you’re sitting in it)
- Creates exception in immunity provided by CDA 230

Allows legal case to be brought against computer service when content is algorithmically amplified:
“the claim involves a case in which the interactive computer service used an algorithm, model, or other computational process to rank, order, promote, recommend, amplify, or similarly alter the delivery or display of information (including any text, image, audio, or video post, page, group, account, channel, or affiliation) provided to a user of the service if the information is directly relevant to the claim.”

H.R. 2154 provides caveat to notion of algorithmic amplification: “the requirement is not met if—
(I) the information delivery or display is ranked, ordered, promoted, recommended, amplified, or similarly altered in a way that is obvious, understandable, and transparent to a reasonable user based only on the delivery or display of the information (without the need to reference the terms of service or any other agreement), including sorting information—
   (aa) chronologically or reverse chronologically;
   (bb) by average user rating or number of user reviews;
   (cc) alphabetically;
   (dd) randomly; and
   (ee) by views, downloads, or a similar usage metric; or
(II) the algorithm, model, or other computational process is used for information a user specifically searches for.”

Is this a reasonable path forward?
Does Section 230 protect AI-generated text and images?

- Section 230 inoculates platforms from liability for content hosted on them
- But now platforms aren’t just hosting content, they are generating it
- Example: Sydney, Bing’s Search AI; ChatGPT

So, will companies like Microsoft be held liable for the content they produce?


“Artificial intelligence generates poetry... It generates polemics today that would be content that goes beyond picking, choosing, analyzing or digesting content. **And that is not protected.** Let’s assume that’s right. Then the question becomes, what do we do about recommendations?”