Computers, Ethics, and Public Policy

Stanford | Engineering
Computer Science

Stanford | McCoy Family Center for Ethics In Society

Stanford | Political Science
Assignment #4 will be available later tonight on the CS181 website and is due Friday, March 15th

Please sign up for group chats with next week’s speakers: Krishna Bharat (founder, Google News), Marietje Schaake (European Parliament), Alex Stamos (former CSO, Facebook)

- Sign-up available on CS198 website
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 information 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 information bubbles
The Power of Network Effects

• The power of platforms often comes from network effects
  • The more entities in the network, the more useful it is
  • 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:
  
  ![Graph Example 1](image1)
  ![Graph Example 2](image2)

• On the web
  • Elements in $V$ are webpages; elements in $E$ are links between webpages

• In a social network
  • Elements in $V$ are users; elements in $E$ represent are 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 #4: Social network simulation
5. Perspectives on information 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

The fraction of users with degree $k$ for both the global and U.S. population of Facebook users. (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.”

Who is that?
1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #4: Social network simulation
5. Perspectives on information 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
    - 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 you have not rated/bought that others ranked highly

• 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
  - E.g., How often two users click on each others posts
  - E.g., How often a user reads an article suggested from a particular site
  - E.g., 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”

- Open questions: Which interactions to model? How to make 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?

“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 simply: (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

<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 information bubbles
Want to examine the effect that content recommendation in a social network has with respect to political polarization
- Polarization impacts free flow of information, is responsible for “echo chambers”, 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.
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>0.3</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.
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:

<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)
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 displays social network as a graph
  • Users are vertices (circles), color-coded by 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)

• Additional information:
  • Total articles shown (to all users)
  • Total articles read (to all users)
  • Percentage of articles read
  • Total revenue: $0.05 \times (# \text{ read})
Day #: 500

Total articles shown: 150870

Total articles read: 84147

Percentage read: 55.77%

Total revenue: $4207.35
1. Formal representations for networks
2. Network structures
3. Models for content recommendation
4. Assignment #4: Social network simulation
5. Perspectives on information bubbles
Perspectives on Information 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
Flaxman, Goel, and Rao, 2016:
- Pairs of users tend to be more segregated (differ in political slant of news sites read) when getting news from social networks than direct sites.
- But, individual users in social networks read news articles with greater variability of political slant then those reading direct sites.
• Information flow on platforms can be highly influenced by both explicit (censoring) and implicit (optimization) decisions.

• What is required for a well-functioning democracy?
  • Is there an obligation for platforms to support diverse information flow?
  • 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 “fake news”?
  • If so, how is determination of “fakeness” made?
    • E.g., Should machine learning be used to detect “fake news”? If not, what will scale?

• These are questions that technologists will need to be involved in answering, because their code impacts the answers!