Collaborative Filtering

Lecture 15
Information Filtering recap

- Look at each document’s content, and see if it matches user’s interests
- Construct user’s initial profile
  - Use examples of relevant and irrelevant documents
  - Mine content from home page, job descriptions, etc
  - Learn profile using e.g. relevance feedback
- Match incoming documents to profiles
- Refine profile from user feedback

- Examples: TREC filtering task
Reducing Information Overload

- Filtering is a response to information overload
- Content-based filtering
  - “I only want to read about this topic.”
  - “I don’t want to read articles by this person.”
- But people filter based on other features, too
  - reputation, popularity
  - reliability
  - novelty
  - recommendation from friend or colleague
Collaborative Filtering

- Filtering based on recommendations
  - Find people with similar taste to yours
  - Recommend to you what they’ve liked
- Automating “word-of-mouth”
  - Computers can consider thousands of items
- Lazy-evaluation filtering
  - Don’t examine content directly
  - Wait until someone else recommends it
Why Collaborative Filtering?

• Many people are watching the same stream
  • Some of them may have overlapping interests
    • e.g. “US-China relations”, “air accidents”, “international border disputes”
  • Leverage group effort
• Recommendations capture what content can’t
  • quality
  • preference
  • popularity
  • utility
How to filter collaboratively

- Collect ratings of documents from users
  - thumbs up/down
  - five-point response scale
- Identify users who have similar tastes
  - compute correlations of users' ratings
  - user-user similarity measure
- Use the most similar users to predict future ratings
## Movie Ratings

<table>
<thead>
<tr>
<th></th>
<th>Comedies</th>
<th></th>
<th>Dramas</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>3</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>?</td>
<td>9</td>
<td>?</td>
<td>2</td>
</tr>
<tr>
<td>Craig</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Diane</td>
<td>?</td>
<td>9</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Lecture 15  
Information Retrieval
How to do it

- User ratings form a ratings matrix
  - users $\times$ documents, where elements are ratings
- Compute correlation coefficients

$$ r(u_1, u_2) = \frac{(u_1 - \overline{u}) \cdot (u_2 - \overline{u})}{\|u_1 - \overline{u}\| \|u_2 - \overline{u}\|} $$

- Predicted rating = weighted sum of (rating * correlation coefficient)
How to do it (Part 2)

<table>
<thead>
<tr>
<th></th>
<th>Comedies</th>
<th></th>
<th>Dramas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>3 5</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Bob</td>
<td>4 9</td>
<td></td>
<td>7 9 2</td>
</tr>
<tr>
<td>Craig</td>
<td>4 9</td>
<td></td>
<td>7 8 1</td>
</tr>
<tr>
<td>Damian</td>
<td>4 9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lecture 15 Information Retrieval
Tapestry (Xerox PARC, ’92)

- Mail and document filtering system
- Both content-based and collaborative
  - users can annotate (endorse) documents
  - filtering on standard fields, as well as ‘replied-to-by’, ‘endorsed-by’
- TQL: an SQL-like, persistent query language
Ringo (MIT, ’95)

- **Goal:** recommend music albums
- **Accessible via email or WWW**
- **Prediction algorithm**
  - Initial profile: ratings (1-7 scale) for 125 artists
  - Predictions computed using correlation method
- **Interface**
  - Recommendation included confidence measure
  - Could suggest artists/albums you would like or hate
  - Could advise regarding a specific album
  - Users can write textual reviews, add artists & albums
GroupLens (U. Minnesota, ’94)

- **Goal:** recommend USENET articles
- **Problem:** USENET is distributed!
  - collect ratings at central servers
  - client connects to both NNTP server and GLRB
- **Problem:** People like their newsreaders
  - augmented existing readers: xrn, tin, gnus
  - worked to make rating a “minimally intrusive” task
  - goal for user is to spend less time reading news
GroupLens Design Goals

- Integrate with existing newsreaders
- Provide fast, current recommendations
  - ~24 predictions/sec (SPARC 5, 32MB, 1997)
  - single key rating input
  - articles are read quickly, then expire
  - 1-day delay in predictions means missing half the users
- Privacy
  - uses pseudonyms with an authentication protocol
Ratings in GroupLens

![Graphs showing ratings distribution for different groups](image-url)
Correlation Between Users
Other GroupLens Applications

- MovieLens: http://movielens.umn.edu/
- BeerLens?
- NetPerceptions
  - Amazon.com, CDnow, Wine.com
Yenta: another kind of CF

- A matchmaking tool
  - clusters documents on your computer
  - characterizes clusters by key words
  - finds other people with similar clusters
- Finds people, not documents
  - locate experts and others who share interests
ReferralWeb (AT&T Labs, ’96)

- Constructing and searching social networks
  - “Find me a friend of a friend who knows about collaborative filtering.”
- Works like a Web spider
  - Searches for documents which mention a person
    - web pages, technical papers, USENET threads, organizational charts
  - Finds names on those pages and searches for them
  - Recurses one or two levels
PHOAKS (AT&T Labs, ’96)

• “People Helping One Another Know Stuff”
• Automatically extracts URLs from USENET articles
  • mentioning URL == recommendation, if
  • message is not crossposted to many groups
  • URL not in .signature
  • URL not in quoted portion of article
  • If it doesn’t look like an advertisement
• Compiles the essential URLs for newsgroups
Pros and Cons of CF

- Can recommend
  - books, movies, CDs, ... anything that can be rated
  - based on quality, authority, popularity, utility...

- Limitations
  - can only recommend what has been rated
  - can only recommend to someone who rates
    - sparse ratings or cold start problem
  - Lots of computation and storage