Collaborative Filtering

Luo Si
Department of Computer Science
Purdue University

Abstract

Outline
- Introduction to collaborative filtering
- Main framework
- Memory-based collaborative filtering approach
- Model-based collaborative filtering approach
  - Aspect model & Two-way clustering model
  - Flexible mixture model
  - Decouple model
- Unified filtering by combining content and collaborative filtering

What is Collaborative Filtering?

Collaborative Filtering (CF): Making recommendation decisions for a specific user based on the judgments of users with similar tastes

| Train_User 1 | 1 | 5 | 3 | 3 | 4 |
| Train_User 2 | 4 | 1 | 5 | 3 | 2 |
| Test User    | 1 | ? | 3 | 4 |

Why Collaborative Filtering?

- Advantages of Collaborative Filtering
  - Collaborative Filtering does not need content information as required by CBF
  - The contents of items belong to the third-party (not accessible or available)
  - The contents of items are difficult to index or analyze (e.g., multimedia information)

- Problems of Collaborative Filtering
  - Privacy issues, how to share one’s interest without disclosing too much detailed information?
Why Collaborative Filtering?

- Applications Collaborative Filtering
  - E-Commerce
  - Email ranking: borrow email ranking from your office mates (be careful...)
  - Web search? (e.g., local search)

Formal Framework for Collaborative Filtering

<table>
<thead>
<tr>
<th>Training Users: U</th>
<th>Objects: O</th>
<th>What we have:</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>O1 : 3 2 4</td>
<td>• Assume there are some ratings by training users</td>
</tr>
<tr>
<td>U2</td>
<td>O2 : 4 1 1</td>
<td>• Test user provides some amount of additional training data</td>
</tr>
<tr>
<td>U3</td>
<td>O3 : 1 1 1</td>
<td></td>
</tr>
<tr>
<td>U4</td>
<td>O4 : 2 2 2</td>
<td></td>
</tr>
<tr>
<td>Test User U5</td>
<td>O5 : 2 3 3</td>
<td></td>
</tr>
</tbody>
</table>

What we do:
- Predict test user's rating based training information

Memory-Based Approaches

- How to determine the similarity between users?
  - Measure the similarity in rating patterns between different users

Pearson Correlation Coefficient Similarity

\[ w_{ui} = \frac{\sum (R_{ui} - \bar{R}_u)(R_{oi} - \bar{R}_o)}{\sqrt{\sum (R_{ui} - \bar{R}_u)^2 \sum (R_{oi} - \bar{R}_o)^2}} \]

Vector Space Similarity

\[ w_{ui} = \frac{\sum R_{ui} R_{oi}}{\sqrt{\sum R_{ui}^2 \sum R_{oi}^2}} \]

Average Ratings

\[ \bar{R}_o = \frac{\sum R_{oi}}{m} \]

Prediction:

\[ \hat{R}_{ui} = \bar{R}_u + \sum w_{ui} (R_{oi} - \bar{R}_o) \]

Memory-Based Approaches

- How to combine the ratings from similar users for predicting?
  - Weight similar users by their similarity with a specific user; use these weights to combine their ratings.

Prediction:

\[ \hat{R}_{ui} = \bar{R}_u + \sum w_{ui} (R_{oi} - \bar{R}_o) \]

Memory-Based Approaches

- Memory-Based Approaches
  - Given a specific user \( u \), find a set of similar users
  - Predict \( u \)'s rating based on ratings of similar users

Issues

- How to determine the similarity between users?
- How to combine the ratings from similar users to make the predictions (how to weight different users?)

Memory-Based Approaches

Train User 1: 1 5 3 3 4
Train User 2: 4 1 5 3 2
Test User: 1 \[ \square \] 3 2 4

Remove User-specific Rating Bias
### Memory-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train User 1</strong></td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Sub Mean (Train1)</strong></td>
<td>-2.2</td>
<td>1.8</td>
<td>-0.2</td>
<td>-0.2</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>Train User 2</strong></td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Sub Mean (Train2)</strong></td>
<td>1</td>
<td>-2</td>
<td>2</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td><strong>Test User</strong></td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Sub Mean (Test)</strong></td>
<td>-1.667</td>
<td>0.333</td>
<td>1.33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Normalize Rating

### Memory-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train User 1</strong></td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Sub Mean (Train1)</strong></td>
<td>-2.2</td>
<td>1.8</td>
<td>-0.2</td>
<td>-0.2</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>Train User 2</strong></td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Sub Mean (Train2)</strong></td>
<td>1</td>
<td>-2</td>
<td>2</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td><strong>Test User</strong></td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Sub Mean (Test)</strong></td>
<td>-1.667</td>
<td>0.333</td>
<td>1.33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Make Prediction:**

\[
2.67 + \frac{1.8 \times 0.92 + (-2) \times (-0.44)}{0.92 + 0.44} = 4.53
\]

### Memory-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train User 1</strong></td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Sub Mean (Train1)</strong></td>
<td>-2.2</td>
<td>1.8</td>
<td>-0.2</td>
<td>-0.2</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>Train User 2</strong></td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Sub Mean (Train2)</strong></td>
<td>1</td>
<td>-2</td>
<td>2</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td><strong>Test User</strong></td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Sub Mean (Test)</strong></td>
<td>-1.667</td>
<td>0.333</td>
<td>1.33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calculate Similarity: \( W_{\text{trn1, test}} = 0.92; W_{\text{trn2, test}} = -0.44 \)

### Memory-Based Approaches

- **Problems with memory-based approaches**
  - Associated a large amount of computation online costs (have to go over all users, any fast indexing approach?)
  - Heuristic method to calculate user similarity and make user rating prediction

- **Possible Solution**
  - Cluster users/items in offline manner, save for online computation cost
  - Proposal more solid probabilistic modeling method

### Collaborative Filtering

**Model-Based Approaches:**

- **Aspect Model (Hofmann et al., 1999)**
  - Model individual ratings as convex combination of preference factors
  
  \[
  P(o_j | u_i, Z) = \sum_{l \in Z} P(o_j | v_l) P(v_l | Z) P(Z)
  \]

- **Two-Sided Clustering Model (Hofmann et al., 1999)**
  - Assume each user and item belong to one user and item group.
  
  \[
  P(o_j, u_i, Z) = P(u_i) P(v_l) \sum_{l \in Z} P(v_l | Z) C_{u_i, v_l}
  \]

  \( C_{u_i, v_l} \): Indicator Variables
  \( Z \): Association Parameter
Thoughts:

Previous algorithms all cluster users and objects either implicitly (memory-based) or explicitly (model-based)

- Aspect model allows users and objects to belong to different classes, but cluster them together
- Two-sided clustering model clusters users and objects separately, but only allows them to belong to one single class

Collaborative Filtering

Flexible Mixture Model (FMM):
Cluster users and objects separately AND allow them to belong to different classes

\[
P(z_u, z_o, r_{u|o}) = \sum_{z_u} P(z_u) P(z_o) P(r_{u|o} | z_u, z_o)\]

- Training Procedure:
  - Annealed Expectation Maximization (AEM) algorithm
  - E-Step: Calculate Posterior Probabilities

Decoupled Model (DM):
Separate preference value

\[
Z_{uo} \in \{1, \ldots, k\} \quad \{1 \text{ disfavor, } k \text{ favor} \}
\]

Joint Probability:

\[
P(z_u, z_o, r_{u|o}) = \sum_{z_u} P(z_u) P(z_o) P(r_{u|o} | z_u, z_o)\]

"Preference-Based Graphical Model for Collaborative Filtering", UAI'03

"A study of Mixture Model for Collaborative Filtering", Journal of IR
**Experimental Data**

Datasets:
- MovieRating and EachMovie

<table>
<thead>
<tr>
<th></th>
<th>MovieRating</th>
<th>EachMovie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Users</td>
<td>500</td>
<td>2000</td>
</tr>
<tr>
<td>Number of Movies</td>
<td>1000</td>
<td>1682</td>
</tr>
<tr>
<td>Avg. # of rated items/User</td>
<td>87.7</td>
<td>129.6</td>
</tr>
<tr>
<td>Scale of ratings</td>
<td>1, 2, 3, 4, 5, 6</td>
<td>1, 2, 3, 4, 5, 6</td>
</tr>
</tbody>
</table>

Evaluation:
MAE average absolute deviation of the predicted ratings to the actual ratings on items.

\[ MAE = \frac{1}{L} \sum_{i=1}^{L} |\hat{r}_{ij} - R_{ij}| \]

**Combine Collaborative Filtering and Content-Based Filtering**

**Content-Based Filtering (CBF):** Recommend by analyzing the content information
- A group of aliens visit earth
- Science Fiction?
  - Yes

**Young Harry is in love and wants to marry an actress, much to the displeasure of his family.**
- No

**Unified Filtering (UF):** Combining both the content-based information and the collaborative rating information for more accurate recommendation

**Collaborative Filtering**

Vary Number of Training User
- For MovieRating
  - 100 and 200 training users
- For EachMovie
  - 200 and 400 training users

Vary Amount of Given Information from the Test User
- For both testbeds
  - Vary among given 5, 10, or 20 items

**Unified Filtering by flexible mixture model and exponential model**

Unified Filtering with mixture model and exponential model (UFME):

Mixture model for rating information:

\[ P(Z_u | U_l) = \sum_{x,y} P(Z_u | Z_x, Z_y) P(\alpha(x,y)) P(\alpha) \]

Exponential model for content information

\[ P(O) = \prod_{l=1}^{L} P(o_l) \]

\[ P(U) = \prod_{l=1}^{L} P(u_l) \]

\[ P(Z_u | U_l) = \prod_{l=1}^{L} P(Z_u | Z_x, Z_y) \]

**Experiment Results**

Table. MAE results for four filtering algorithms on EachMovie testbed. Four algorithms are pure content-based filtering (CBF), pure collaborative filtering (CF), unified filtering by combining mixture model and exponential model (UFME)

<table>
<thead>
<tr>
<th>Training Users Size</th>
<th>Algorithms</th>
<th>0 Items Given</th>
<th>5 Items Given</th>
<th>10 Items Given</th>
<th>20 Items Given</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>CBF</td>
<td>1.43</td>
<td>1.21</td>
<td>1.24</td>
<td>1.19</td>
</tr>
<tr>
<td></td>
<td>CF</td>
<td>1.21</td>
<td>1.14</td>
<td>1.13</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>UFME</td>
<td>1.19</td>
<td>1.11</td>
<td>1.10</td>
<td>1.09</td>
</tr>
<tr>
<td>100</td>
<td>CBF</td>
<td>1.43</td>
<td>1.23</td>
<td>1.21</td>
<td>1.19</td>
</tr>
<tr>
<td></td>
<td>CF</td>
<td>1.17</td>
<td>1.08</td>
<td>1.07</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>UFME</td>
<td>1.17</td>
<td>1.08</td>
<td>1.06</td>
<td>1.05</td>
</tr>
</tbody>
</table>

**Results of Flexible Mixture Model**

[Graphs showing experiment results]
Experiment Results

The table below lists the five most indicative words (with highest values) for 5 movie clusters, sorted by each column corresponds to a different movie cluster. All listed words are stemmed.

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>forev</td>
<td>previou</td>
<td>mad</td>
<td>inhabit</td>
<td>custom</td>
</tr>
<tr>
<td>depress</td>
<td>passion</td>
<td>hang</td>
<td>dress</td>
<td>hang</td>
</tr>
<tr>
<td>mate</td>
<td>court</td>
<td>rape</td>
<td>relat</td>
<td>forev</td>
</tr>
<tr>
<td>broken</td>
<td>forget</td>
<td>finish</td>
<td>door</td>
<td>water</td>
</tr>
<tr>
<td>abandon</td>
<td>sea</td>
<td>arrest</td>
<td>younger</td>
<td>food</td>
</tr>
</tbody>
</table>

Abstract

Outline
- Introduction to collaborative filtering
- Main framework
- Memory-based collaborative filtering approach
- Model-based collaborative filtering approach
  - Aspect model & Two-way clustering model
  - Flexible mixture model
  - Decouple model
- Unified filtering by combining content and collaborative filtering