Collaborative Filtering

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

Content-Based Filtering: Recommend by analyzing the content information

Collaborative Filtering: Make recommendation by judgments of similar users

<table>
<thead>
<tr>
<th></th>
<th>Train_User 1</th>
<th>Train_User 2</th>
<th>Test_User</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
What is Collaborative Filtering?

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

<table>
<thead>
<tr>
<th>Train_User 1</th>
<th>1</th>
<th>5</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train_User 2</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Test User</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

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
    - Amazon.com
    - Yahoo Shopping
    - eBay
    - Catalog City
    - Arcade City
    - Buy.com
    - Volatile Music
  - 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>Objects: $O_m$</th>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
<th>……</th>
<th>$O_j$</th>
<th>……</th>
<th>$O_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Users: $U_n$</td>
<td>$U_1$</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$U_2$</td>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$U_i$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$U_N$</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test User $U_t$</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What we have:
- Assume there are some ratings by training users
- Test user provides some amount of additional training data

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

$$R_{ui} (O_j) = ?$$
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**

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

**Pearson Correlation Coefficient Similarity**

$$W_{u,u'} = \sum \frac{(R_u(o) - \bar{R}_u)(R_{u'}(o) - \bar{R}_{u'})}{\sqrt{\sum (R_u(o) - \bar{R}_u)^2} \sqrt{\sum (R_{u'}(o) - \bar{R}_{u'})^2}}$$

**Vector Space Similarity**

$$W_{u,u'} = \sum \frac{R_{u'}(o)R_u(o)}{\sqrt{\sum R_{u'}(o)^2} \sqrt{\sum R_u(o)^2}}$$

**Average Ratings**

$$\bar{R}_u(o) = \bar{R}_{u'} + \frac{\sum W_{u,u'} (R_u(o) - \bar{R}_u)}{\sum W_{u,u'}}$$

**Prediction:**

$$R_{u'}(o) = \bar{R}_{u'} + \frac{\sum W_{u,u'} (R_u(o) - \bar{R}_u)}{\sum W_{u,u'}}$$
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:

\[
R_{u'}(o) = \frac{\sum_u w_{u,u'} (R_u(o) - \bar{R}_u)}{\sum_u |w_{u,u'}|}
\]

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

<table>
<thead>
<tr>
<th></th>
<th>Rating 1</th>
<th>Rating 2</th>
<th>Rating 3</th>
<th>Rating 4</th>
<th>Rating 5</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>?</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

Calculate Similarity: $W_{trn1\_test}=0.92; W_{trn2\_test}=0.44$
### Memory-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>5</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>?</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: \[
2.67 + (1.8 \times 0.92 + (-2) \times (-0.44)) / (0.92 + 0.44) = 4.54
\]
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

\[
P(o_{ij}, u_j, r_{ij}) = \sum_{z \in Z} P(z) P(o_{ij} | z) P(u_j | z) P(r_{ij} | z)
\]

Two-Sided Clustering Model (Hofmann et al., 1999)
- Assume each user and item belong to one user and item group.

\[
P(o_{ij}, u_j, r_{ij}) = P(o_{ij}) P(u_j) \sum_{y \in y} \sum_{x \in x} I_{x(l)} J_{y(l)} C_{xy} \tag{1}
\]

Brief description for the expectation maximization training process…
Collaborative Filtering

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

Previous Work: Thoughts

Cluster users and objects separately AND allow them to belong to different classes

Flexible Mixture Model (FMM)
Collaborative Filtering

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

\[ P(a_{(i)}, u_{(i)}, r_{(i)}) = \sum_{Z_u, Z_o} P(Z_o) P(Z_u) P(a_{(i)} | Z_o) P(u_{(i)} | Z_u) P(r_{(i)} | Z_u, Z_o) \]

• Training Procedure:
  Annealed Expectation Maximization (AEM) algorithm

  E-Step: Calculate Posterior Probabilities

\[ P(z_o, z_u | a_{(i)}, u_{(i)}, r_{(i)}) = \frac{(P(Z_o) P(Z_u) P(a_{(i)} | Z_o) P(u_{(i)} | Z_u) P(r_{(i)} | Z_u, Z_o))^k}{\sum_{Z_o, Z_u} (P(Z_o) P(Z_u) P(a_{(i)} | Z_o) P(u_{(i)} | Z_u) P(r_{(i)} | Z_u, Z_o))^k} \]

M-Step: Update Parameters

• Prediction Procedure:
  Fold-In process to calculate join probabilities

\[ P(o, u', r_{(i)}) = \sum_{Z_o, Z_u} P(Z_o) P(Z_u) P(o | Z_o) P(u' | Z_u) P(r | Z_o, Z_u) \]

Fold-in process by EM algorithm

Calculate expectation for prediction

\[ \hat{R}_{u'}(o) = \sum_r \sum_{r'} P(o, u', r) \cdot \frac{P(o, u', r)}{\sum_r \sum_{r'} P(o, u', r')} \]

“Flexible Mixture Model for Collaborative Filtering”, ICML’03
Collaborative Filtering

Thoughts:
• Previous algorithms address the problem that users with similar tastes may have different rating patterns implicitly (Normalize user rating)

Previous Work: Thoughts

• Thoughts:

<table>
<thead>
<tr>
<th>Ms. Nice</th>
<th>Mr. Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nice Rating: 5</td>
<td>Nice Rating: 3</td>
</tr>
<tr>
<td>Mean Rating: 2</td>
<td>Mean Rating: 1</td>
</tr>
</tbody>
</table>

Explicitly decouple users preference values out of the rating values
Decoupled Model (DM)
Decoupled Model (DM)

- **Decoupled Model (DM):**
  Separate preference value
  
  \[ Z_{\text{pref}} \in [1, \ldots, k] \]  (1 disfavor, k favor)
  
  from rating  \( r \in \{1, 2, 3, 4, 5\} \)

**Joint Probability:**

\[
P(o_{(i)}, u_{(i)}, r_{(i)}) = \sum_{Z_u, Z_o} P(Z_u) P(o_{(i)} | Z_u) P(u_{(i)} | Z_u) P(Z_o | u_{(i)}) \sum_{r_{(j)}} P(Z_{\text{pref}} | Z_u, Z_o) P(r_{(j)} | Z_{\text{pref}}, Z_u) \]

"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</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.

\[
\text{MAE} = \frac{1}{L_{\text{test}}} \sum_{l} |r_{(i)} - \hat{R}_{\text{test}}(u_{(i)})| \]
Collaborative Filtering

Vary Number of Training Users

Test behaviors of algorithms with different amount of training data
- For MovieRating
  100 and 200 training users
- For EachMovie
  200 and 400 training users

Vary Amount of Given Information from the Test User

Test behaviors of algorithms with different amount of given information from test user
- For both testbeds
  Vary among given 5, 10, or 20 items

Results of Flexible Mixture Model
## Experimental Results

Improved by Combing FMM and DM

<table>
<thead>
<tr>
<th>Training Users Size</th>
<th>Algorithms</th>
<th>5 Items Given</th>
<th>10 Items Given</th>
<th>20 Items Given</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>FMM</td>
<td>0.829</td>
<td>0.822</td>
<td>0.807</td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>0.792</td>
<td>0.772</td>
<td>0.741</td>
</tr>
<tr>
<td>200</td>
<td>FMM</td>
<td>0.800</td>
<td>0.787</td>
<td>0.768</td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>0.770</td>
<td>0.750</td>
<td>0.728</td>
</tr>
<tr>
<td>200</td>
<td>FMM</td>
<td>1.07</td>
<td>1.04</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>1.06</td>
<td>1.01</td>
<td>0.99</td>
</tr>
<tr>
<td>400</td>
<td>FMM</td>
<td>1.05</td>
<td>1.03</td>
<td>1.01</td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>1.04</td>
<td>1.00</td>
<td>0.97</td>
</tr>
</tbody>
</table>

### Results on Movie Rating

### Results on Each Movie

---

## Combine Collaborative Filtering and Content-Based Filtering

**Content-Based Filtering (CBF):** Recommend by analyzing the content information

Content information is very useful when few users have rated an object.

A group of **aliens** visit earth..........................  
kind of friendship in which **E.T** learns..............  
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
Content-Based Filtering and Unified Filtering

Content-Based Filtering (CF):

- Generative Methods (e.g. Naïve Bayes)
- Discriminative Methods (e.g. SVM, Logistic Regression)
  - Usually more accurate
  - Can be used to combine features (e.g., actors for movies)

Unified Filtering by combining CF and CBF:

- Linearly combine the scores from CF and CBF
- Personalized linear combination of the scores
- Bayesian combination with collaborative ensemble learning

Unified Filtering by flexible mixture model and exponential model

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

Mixture model for rating information:

\[ P(o_i, u_i, r_{ij}) = \sum_{z_u, z_o} P(z_o | \bar{d}_o) P(z_u | u_i) P(o_i) P(u_i) P(r_{ij} | z_u, z_o) \]

Exponential model for content information
Unified Filtering by flexible mixture model and exponential model

- Unified Filtering with mixture model and exponential model (UFME):
  - Mixture model for rating information:
  - Exponential model for content information:

\[
P_{\theta}(Z_o \mid d_{ol}) = \frac{\exp(\sum_j \theta_{z_o,j} \ast d_{oj})}{\sum_{z_o} \left( \exp(\sum_j \theta_{z_o,j} \ast d_{oj}) \right)}
\]

Specific word

Training Procedure:

- E-Step: Calculate posterior probabilities
  - Expectation Step of EM
- M-Step: Update parameters
  - Second, refine the object cluster distribution with content information by maximizing Iterative Scaling Training

“Unified Filtering by Combining Collaborative Filtering and Content-Based Filtering via Mixture Model and Exponential Model ”, CIKM'04
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>

Experiment Results

\[ P_\theta(Z_w | w) \]

Table. 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>
Summary

What we talked about so far?

• Proposed the flexible mixture model
  – Demonstrates the power of clustering users and objects separately AND allowing them to belong to different classes

• Proposed the decoupled model
  – Demonstrates the power of extracting preference values from the surface rating values

• Proposed the unified probabilistic model for unified filtering
  – Demonstrates the power of taking advantage of content information with limited rating information