Collaborative Filtering

Prof. Chris Clifton

17 October 2019

Material adapted from course created by Dr. Luo Si, now leading Alibaba research group

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

Train_User 1
1 5 3 3 4
Train_User 2
4 1 5 3 2
Test User
1 ? 3 4
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></th>
<th>Train_User 1</th>
<th>Train_User 2</th>
<th>Test User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Romantic</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Star Wars II</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>A.I.</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Triangle</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>E.T.</td>
<td>4</td>
<td>2</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

- 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: $U_n$</th>
<th>Test User $U_t$</th>
<th>Objects: $O_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U_1$</td>
<td>$O_1$</td>
</tr>
<tr>
<td></td>
<td>$U_2$</td>
<td>$O_2$</td>
</tr>
<tr>
<td></td>
<td>$U_i$</td>
<td>$O_3$</td>
</tr>
<tr>
<td></td>
<td>$U_N$</td>
<td>$O_j$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$O_M$</td>
</tr>
<tr>
<td>$3$</td>
<td>$2$</td>
<td>$4$</td>
</tr>
<tr>
<td>$4$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>$5$</td>
<td>$2$</td>
<td>$2$</td>
</tr>
<tr>
<td>$2$</td>
<td>$3$</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

$R_{ut}(O_j) =$ ?
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**
  \[
  w_{u,u'} = \frac{\sum (R_u(o) - \bar{R}_u)(R_{u'}(o) - \bar{R}_{u'})}{\sqrt{\sum (R_u(o) - \bar{R}_u)^2 \sum (R_{u'}(o) - \bar{R}_{u'})^2}}
  \]

  **Vector Space Similarity**
  \[
  w_{u,u'} = \frac{\sum R_{u'}(o)R_u(o)}{\sqrt{\sum R_{u'}(o)^2 \sum R_u(o)^2}}
  \]

- 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.

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

Memory-Based Approaches

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

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

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

Normalize Rating

### Memory-Based Approaches

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

Calculate Similarity: Wtrn1_test=0.92; Wtrn2_test=-0.44;
## Memory-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>Train_User 1</th>
<th>Train_User 2</th>
<th>Test User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train_User 1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Sub Mean (Train1)</td>
<td>-2.2</td>
<td>1</td>
<td>1.667</td>
</tr>
<tr>
<td>Sub Mean (Train2)</td>
<td>1</td>
<td>-2</td>
<td>5</td>
</tr>
<tr>
<td>Sub Mean (Test)</td>
<td>-1.667</td>
<td>0.333</td>
<td>4</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

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

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

- Training Procedure:
  Annealed Expectation Maximization (AEM) algorithm

E-Step: Calculate Posterior Probabilities

\[
P(z_o, z_u | a_{ij}, u_{ij}, r_{ij}) = \frac{(P(Z_o)P(Z_u)P(a_{ij} | Z_o)P(u_{ij} | Z_u)P(r_{ij} | Z_o, Z_u))^k}{\sum_{Z_o, Z_u} (P(Z_o)P(Z_u)P(a_{ij} | Z_o)P(u_{ij} | Z_u)P(r_{ij} | Z_o, Z_u))^k}
\]
\[ P(Z_o); P(Z_u); P(o_{(l)} \mid Z_o); P(u_{(l)} \mid Z_u); P(r_{(l)} \mid Z_o, Z_u) \]

M-Step: Update Parameters

- Prediction Procedure:
  Fold-In process to calculate joint probabilities

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

Fold-in process by EM algorithm

Calculate expectation for prediction

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

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

Thoughts:

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

• Thoughts:

Explicitly decouple users preference values out of the rating values

Decoupled Model (DM)

Decoupled Model (DM):
Separate preference value

\[ Z_{\text{pref}} \in \{1,\ldots,k\} \quad (1 \text{ disfavor}, k \text{ favor}) \]

from rating \( r \in \{1,2,3,4,5\} \)

Joint Probability:

\[
P(o_{(i)}, u_{(i)}, r_{(i)}) = \sum_{Z_o, Z_u, Z_k} P(Z_o) P(Z_u) P(o_{(i)} | Z_o) P(u_{(i)} | Z_u) P(r_{(i)} | Z_k) \sum_{Z_{ov}} P(Z_{ov} | Z_o, Z_u) P(r_{(i)} | Z_{ov}, Z_k)\]

"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.

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

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
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>
<th>Results on Movie Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>FMM</td>
<td>0.829</td>
<td>0.822</td>
<td>0.807</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>0.792</td>
<td>0.772</td>
<td>0.741</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>FMM</td>
<td>0.800</td>
<td>0.787</td>
<td>0.768</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>0.770</td>
<td>0.750</td>
<td>0.728</td>
<td></td>
</tr>
</tbody>
</table>

<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>
<th>Results on Each Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>FMM</td>
<td>1.07</td>
<td>1.04</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>1.06</td>
<td>1.01</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>FMM</td>
<td>1.05</td>
<td>1.03</td>
<td>1.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMM+DM</td>
<td>1.04</td>
<td>1.00</td>
<td>0.97</td>
<td></td>
</tr>
</tbody>
</table>

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.............................. Science Fiction? Yes
- kind of friendship in which E.T learns....................
- 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:

\[
P(\theta_{ij}, u(i), r_{ij}) = \sum_{Z_u, Z_o} P(Z_u | \vec{d}_o) P(Z_o | u) P(\theta_{ij}) P(u_{ij}) P(r_{ij} | Z_o, Z_u)
\]

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:

  \[ P(\mathbf{Z}_u | \mathbf{U}) \]

  \[ P(\mathbf{o}) \]

  \[ P(\mathbf{u}) \]

  \[ Z_0 \]

  \[ d_{olj} \]

  \[ \exp( \sum_j \theta_{z_{u,j}} * d_{olj} ) \]

  \[ \sum_{Z_0} (\exp( \sum_j \theta_{z_{u,j}} * d_{olj} )) \]

  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

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

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