Iterative Algorithms for Collaborative Filtering with Mixture Models
Chalmers University of Technology

Eszter P. Windhager, Libertad Tansini, Istvan Biro & Devdatt Dubhashi

July 7, 2006
Outline

1. Introduction
   - Motivation
   - Mixture Model

2. Methods

3. Results
   - Tests
   - Generated Data
   - Real Data

4. Conclusions

5. Future Work
Outline

1. Introduction
   - Motivation
     - Mixture Model
2. Methods
3. Results
   - Tests
   - Generated Data
   - Real Data
4. Conclusions
5. Future Work
Motivation

- Millions of items to choose from
- Locate reliable experts?
- From individual to collective method of recommendation
  *Collaborative Filtering* – CF
- Amazon’s purchase recommendations
Outline

1 Introduction
   - Motivation
   - Mixture Model

2 Methods

3 Results
   - Tests
   - Generated Data
   - Real Data

4 Conclusions

5 Future Work
Hofmann and Puzicha proposed a probabilistic model
Hidden or latent structure in underlying data
Items are grouped into topics – may overlap arbitrarily
Kleinberg and Sandler gave first theoretical analysis of algorithms for CF in this setting
Mixture Model II

Users

| 1 | 2 | 3 | 4 | 5 |

Items

- cluster1
- cluster2
- cluster3

Windhager, Tansini, Biro & Dubhashi
Iterative Algorithms for Collaborative Filtering - IIA 2006
Mixture Model III

- $n$ users and $m$ items
- Underlying hidden structure of $C$ clusters or topics of items
- Each user $u$ distribution over clusters preference for cluster $c$ is $p(u, c)$ – interest in several clusters
- Each cluster $c$ probability distribution $w(a, c)$ over items $a$ – overlapping
Mixture Model VI

\[ W \cdot P \]

\[ C \]

\[ m \]

\[ n \]
Selection matrix $S$, where $s(u, a) = 1$ if item $a$ was selected by user $u$, $s(u, a) = 0$ otherwise

Suggest items to users

With known $P$ and $W$ matrices – suggest the item $a$ to user $u$ with maximum value in $u$th column of $W \cdot P$ matrix

Try to approximate $P$ and $W$
Objective

- Suit of 4 algorithms tailored to the mixture model
- Simple – conceptually and implementation
- Efficient
Methods

1. Similarity based soft clustering – Pearson correlation – $w_0(a, c)$
2. Given $w_0(a, c)$ compute $p_0(u, c) \forall u, c$ based on HITS idea
3. Given $w_i(a, c)$ and $p_i(u, c)$ compute intermediate values $\bar{p}_{i+1}(u, c)$ and $\bar{w}_{i+1}(a, c) \forall u, a, c$ based on HITS idea
4. Finally calculate the average
   \[
   p_{i+1}(u, c) = (1 - \theta)p_i(u, c) + \theta\bar{p}_{i+1}(u, c)
   \]
   \[
   w_{i+1}(a, c) = (1 - \theta)w_i(a, c) + \theta\bar{w}_{i+1}(a, c),
   \]
   where $0 \leq \theta \leq 1$ is a parameter of the algorithm
5. Until updates are not significant, less then $\epsilon \geq 0$. 

Windhager, Tansini, Biro & Dubhashi
Iterative Algorithms for Collaborative Filtering - IIA 2006
First method

- Sum weights of items in the cluster selected by user and normalize

\[
\bar{p}_{i+1}(u, c) = \frac{\sum_a w_i(a, c) \cdot s(u, a)}{\sum_{c'} \sum_a w_i(a, c') \cdot s(u, a)}
\]

- Similarly sum preferences of users who selected the item for that cluster and normalize

\[
\bar{w}_{i+1}(a, c) = \frac{\sum_u p_i(u, c) \cdot s(u, a)}{\sum_{a' \in c} \sum_u p_i(u, c) \cdot s(u, a')}
\]
Second Method

Items can belong to several clusters

\[
\hat{p}_{i+1}(u, c) = \frac{1}{S_u} \left( \sum_a w_i(a, c) s(u, a) - \frac{1}{C-1} \sum_{a \in c} \sum_{c_r \neq c} w_i(a, c_r) s(u, a) \right)
\]

- \(S_u\) is number of samples for user \(u\) and \(C\) is number of clusters, obviously \(\hat{p}_{i+1}(u, c) \in [-1, 1] \ \forall i, u, c\).

\[
\hat{w}_{i+1}(a, c) = \frac{1}{S_a} \left( \sum_u p_i(u, c) \cdot s(u, a) - \frac{1}{C-1} \sum_{u \in c} \sum_{c_r \neq c} p_i(u, c_r) \cdot s(u, a) \right)
\]

- Item \(a\) was selected \(S_a\) times and \(u \in c\) means that \(p_i(u, c) \neq 0\)
Second Method

- Linear transform to convert values into $(0, 1)$

\[ p'_{i+1}(u, c) = \frac{\hat{p}_{i+1}(u, c) + 1}{2} \]

- Values are highly concentrated around $1/2$, so we re-scale

\[ \bar{p}_{i+1}(u, c) = \frac{1}{N_u} (p'_{i+1}(u, c) - \frac{m_u}{2}) \]

- Normalizing factor for user $u \frac{1}{N_u}$ so $\sum_c \bar{p}_{i+1}(u, c) = 1$ and $m_u = \min_c p'_{i+1}(u, c)$

- Similarly for $\bar{w}_{i+1}(a, c)$
Too Simple

- Shape of the transform function, horizontal axis $\hat{p}_{i+1}(u, c)$ and vertical axis $\bar{p}_{i+1}(u, c)$
Third Method

\( \hat{p}_{i+1}(u, c) \in [-1, 1] \) values concentrated around zero

1. If no item was selected for cluster \( c \) then \( \bar{p}_{i+1}(u, c) = 0 \)

2. if \( \hat{p}_{i+1}(u, c) \geq 0 \)

\[
\bar{p}_{i+1}(u, c) = \frac{1}{N_u} \left( (1 - \frac{1}{C})(\hat{p}_{i+1}(u, c))^{0.3} + \frac{1}{C} \right)
\]

3. if \( \hat{p}_{i+1}(u, c) < 0 \)

\[
\bar{p}_{i+1}(u, c) = \frac{1}{N_u} \left( (\frac{1}{C})(-\hat{p}_{i+1}(u, c))^{0.2} + \frac{1}{C} \right)
\]
Fourth Method

- Like third method without normalizing factor $1/N_u$
- Re-scaling formula of second method
- Typical spread of preferences and weights is wider
Outline

1 Introduction
   - Motivation
   - Mixture Model

2 Methods

3 Results
   - Tests
   - Generated Data
   - Real Data

4 Conclusions

5 Future Work
• Generated test data
• Real data collected from Hungarian web news cite – similar performance to generated data
• Previous benchmark studies
• "Semi-omniscient" algorithm of Kleinberg and Sandler
Outline

1. Introduction
   - Motivation
   - Mixture Model

2. Methods

3. Results
   - Tests
   - Generated Data
   - Real Data

4. Conclusions

5. Future Work
Generated Test Data

- **Disjoint Model** users like only one cluster, \( p(u, c) = 1 \) for exactly one \( c \) and \( p(u, c') = 0 \) for rest
  Disjoint clusters and random weights for items in clusters

- **Partially Mixed** users like one cluster more, \( p(u, c) = 0.9 \) for exactly one \( c \), and \( p(u, c') = \frac{0.1}{C-1} \) for the rest
  Disjoint clusters and random weights for items in clusters

- **Fully Mixed** items can belong to several clusters and users can be interested in several clusters
  Random preferences and weights
  Probability of non zero preference for 1, 2, 3, ..9 clusters is 0.5, 0.3, 0.15, 0.04, 0.005, 0.003, 0.0015, 0.0004, 0.0001
Disjoint Model

- 1000 users, 300 items and 10 clusters
- 10 or 20 random samples for each user

<table>
<thead>
<tr>
<th></th>
<th>Sample 10</th>
<th>Sample 20</th>
<th>Sample 10</th>
<th>Sample 20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong></td>
<td><strong>θ</strong></td>
<td><strong>F 10</strong></td>
<td><strong>F 20</strong></td>
<td><strong>F 10</strong></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4.7</td>
<td>14.9</td>
<td>6.3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4.0</td>
<td>14.3</td>
<td>5.9</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4.2</td>
<td>14.5</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
<td>4.3</td>
<td>14.6</td>
<td>6.1</td>
</tr>
<tr>
<td>1</td>
<td>0.2</td>
<td>8.7</td>
<td>19.1</td>
<td>8.7</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>4.4</td>
<td>15.0</td>
<td>5.8</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>4.5</td>
<td>15.1</td>
<td>5.8</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
<td>4.4</td>
<td>15.1</td>
<td>5.8</td>
</tr>
</tbody>
</table>
### Partially Mixed Model

<table>
<thead>
<tr>
<th>M</th>
<th>( \theta )</th>
<th>Sample 10</th>
<th>Sample 20</th>
<th>Sample 10</th>
<th>Sample 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4.1</td>
<td>14.2</td>
<td>4.3</td>
<td>14.9</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>1.1</td>
<td>2.2</td>
<td>1.1</td>
<td>2.2</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>1.1</td>
<td>2.2</td>
<td>1.2</td>
<td>2.3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.1</td>
<td>2.2</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Windhager, Tansini, Biro & Dubhashi

Iterative Algorithms for Collaborative Filtering - IIA 2006
## Fully Mixed Model

<table>
<thead>
<tr>
<th></th>
<th>Sample 10</th>
<th>Sample 20</th>
<th>Sample 10</th>
<th>Sample 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>θ</td>
<td>F 10</td>
<td>F 20</td>
<td>F 10</td>
</tr>
<tr>
<td>1</td>
<td>1.1</td>
<td>6.4</td>
<td>1.5</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>5.6</td>
<td>1.3</td>
<td>7.1</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>5.7</td>
<td>1.3</td>
<td>7.1</td>
</tr>
<tr>
<td>2</td>
<td>1.7</td>
<td>4.6</td>
<td>1.6</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>7.7</td>
<td>1.9</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>7.7</td>
<td>1.9</td>
<td>7.7</td>
</tr>
<tr>
<td>3</td>
<td>1.9</td>
<td>7.7</td>
<td>1.9</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>7.7</td>
<td>1.9</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>7.7</td>
<td>1.9</td>
<td>7.7</td>
</tr>
</tbody>
</table>

**Windhager, Tansini, Biro & Dubhashi**

Iterative Algorithms for Collaborative Filtering - I3A 2006
Outline

1. Introduction
   - Motivation
   - Mixture Model

2. Methods

3. Results
   - Tests
   - Generated Data
   - Real Data

4. Conclusions

5. Future Work
Real Data

- Log file of a Hungarian news portal (http://origo.hu)
- 1.07 million users and 368000 items from one month
- First test with 1000 users and 8321 selected items
- Without iteration ($\theta = 0$)
- Similar to results on generated data

<table>
<thead>
<tr>
<th>Method</th>
<th>F10</th>
<th>F20</th>
<th>F30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.13</td>
<td>13.77</td>
<td>18.49</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
<td>1.35</td>
<td>3.18</td>
</tr>
<tr>
<td>3</td>
<td>3.71</td>
<td>8.13</td>
<td>11.15</td>
</tr>
<tr>
<td>4</td>
<td>3.50</td>
<td>7.73</td>
<td>10.71</td>
</tr>
</tbody>
</table>
Conclusions

- Disjoint clusters – method 1 works the best with iterations
- Fully mixed model – methods 3 and 4 with iteration – small resources then method 1 without iterating acceptable
- Partially mixed model – not clear – method 1 without iteration works well
- Open question to explain different performance characteristics by a theoretical analysis
- Real data – method 1 best results, as we expected
Future Work

- Expect real data to behave like fully mixed model – complex
- Tests on larger real data sets and with iteration
- Refined evaluation by looking into results manually
- Include multiple selections
- High-order Markov-chains combined with collaborative filtering – meaningful suggestions
Questions?