Hybrid Recommendation

Peter Brusilovsky
with slides of Danielle Lee
IS2480 Adaptive Information Systems

Three basic recommendation engines

- **Collaborative Filtering**: exploiting other likely-minded community data to derive recommendations
  - Effective, Novel and Serendipitous recommendations
  - Data Sparsity, cold-start problem and ad-hoc users
- **Content-based approach**: relying on product (information) features and textual descriptions
- **Knowledge-based approach**: reasoning on explicit knowledge models from the domain
  - Ability to generate recommendation with a small set of user preference and suggest reasonable recommendations
  - Easy to generate too obvious or boring recommendation and plasticity problems.
- Each engine also have variations
  - Content vs. metadata in CBF
  - Peers vs. friends in CF
Input Data Requirements of Recommendation Techniques

<table>
<thead>
<tr>
<th></th>
<th>User Profile &amp; Contextual Parameters</th>
<th>Community Data</th>
<th>Product Features</th>
<th>Knowledge models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative Filtering</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Content-based</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Knowledge-based</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Different engines and their variations typically use difference sources of data. It could be wise to combine the approaches to use more data.

Hybridization Designs

- **Monolithic Hybridization**
  - Incorporating aspects of several recommendation strategies in one algorithm implementation
- **Parallelized Hybridization**
  - Operating independently of one another and produce separate recommendation lists. Then their output is combined into a final set of recommendations
- **Pipelined Hybridization**
  - Several recommender systems are joined together in a pipeline architecture. The output of one recommender becomes part of the input of the subsequent one.
Monolithic Hybridization

- Built-in modification of recommendation algorithm to exploit different types of input data.
- Feature combination hybrids
- Feature augmentation hybrids
### Example (1)

<table>
<thead>
<tr>
<th>User</th>
<th>Item1</th>
<th>Item2</th>
<th>Item3</th>
<th>Item4</th>
<th>Item5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>User1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>User2</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>User3</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>User4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

### Example (1)

<table>
<thead>
<tr>
<th>Item</th>
<th>Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item1</td>
<td>Romance</td>
</tr>
<tr>
<td>Item2</td>
<td>Mystery</td>
</tr>
<tr>
<td>Item3</td>
<td>Mystery</td>
</tr>
<tr>
<td>Item4</td>
<td>Mystery</td>
</tr>
<tr>
<td>Item5</td>
<td>Fiction</td>
</tr>
</tbody>
</table>

**Legend:** If a user bought mainly books of genre X (two-thirds of the total purchases and at least two books), we say that 'User likes many X books'.
Example (2)

<table>
<thead>
<tr>
<th></th>
<th>R nav</th>
<th>R view</th>
<th>R ctx</th>
<th>R buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>n3, n4</td>
<td>i5</td>
<td>k5</td>
<td>null</td>
</tr>
<tr>
<td>User1</td>
<td>n1, n5</td>
<td>i3, i5</td>
<td>k5</td>
<td>i1</td>
</tr>
<tr>
<td>User2</td>
<td>n3, n4</td>
<td>i3, i5, i7</td>
<td>null</td>
<td>i3</td>
</tr>
<tr>
<td>User3</td>
<td>n2, n3, n4</td>
<td>i2, i4, i5</td>
<td>2, k4</td>
<td>i4</td>
</tr>
</tbody>
</table>

Precedence rules: (R buy, R ctx) - R view - R nav

Example (3)

• Elicitation of user feedback and collaborative filtering
  • *Price should be less than the price for item a.*

Monolithic Hybridization

• Feature augmentation hybrids
Parallelized Hybridization

- Employ several recommenders side by side and employ a specific hybridization technique to aggregate the outputs.
- Mixed Hybrids
- Weighted Hybrids
  - Zanker and Jessenitschnig (2009), Claypool, et al. (1999)
- Switching Hybrids
  - Zanker and Jessenitschnig (2009), van Setten (2005)

Parallelized Hybridization

- Mixed Hybrid: combines results of different recommenders at user interface level
Parallelized Hybridization

- Weighted Hybrids: Combines recommendations by computing weighted sums of their scores
Parallelized Hybridization

Why switching might be better than weighting?

- Switching hybrids
Pipelined Hybridization

- A staged process in which several techniques sequentially build on each other before the final one produces recommendations
- Cascade Hybrids
  - Zanker and Jessenitschnig (2009)
- Meta-level Hybrids
  - Zanker (2008), Pazzani (1999)

Cascade hybrids: based on a sequenced order of techniques.
Pipelined Hybridization

- Meta-Level Hybrids: one recommender builds a model that is exploited by the principal recommender

Hybridization Summary

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>Weight</td>
<td>Mixed</td>
<td>Switch</td>
<td>FC</td>
<td>Cascade</td>
<td>FA</td>
<td>Meta</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FC = Feature Combination, FA = Feature Augmentation, CF = collaborative, CN = content-based, DM = demographic, KB = knowledge-based

- Redundant
- Not possible
- Existing implementation