Recommendation and Information Retrieval: Two sides of the same coin?

Prof.dr.ir. Arjen P. de Vries

arjen@acm.org

CWI, TU Delft, Spinque
• Recommendation Systems
  – Collaborative Filtering (CF)

• Probabilistic approaches
  – Language modelling for Information Retrieval
  – Language modelling for log-based CF
  – Brief: adaptations for rating-based CF

• Vector Space Model (“Back to the Future”)
  – User and item spaces, orthonormal bases and “the spectral theorem”
Recommendation

• Informally:
  – Search for information “without a query”

• Three types:
  – Content-based recommendation
  – Collaborative filtering (CF)
    • Memory-based
    • Model-based
  – Hybrid approaches
Recommendation

• Informally:
  – Search for information “without a query”

• Three types:
  – Content-based recommendation
  – Collaborative filtering
    • Memory-based
    • Model-based
  – Hybrid approaches

Today’s focus!
Paradiso begint eigen platenlabel

Het Amsterdamse poppodium Paradiso begint een eigen platenlabel: de Paradiso Vinyl Club. Hiermee brengt het podium alleen werk uit van beginnende Nederlandse muziekcacts.

Door: Jelmer Luimstra   11 februari 2015, 17:17

Het idee: bands leveren zelf de opnamen en het artwork, en Paradiso brengt hun single uit. De popzaal wil acht keer per jaar een 7-inch single op de markt brengen van een andere band. De soldeer van de single...
First public announcement: I will be moving jobs (and house) to beautiful Nijmegen!!

Proud to be taking up the chair of Information Retrieval, even though I feel also sad to leave behind so many friends in Amsterdam and Utrecht.

Oh!!! Nijmegen is REALLY CLOSE BY people!!

Arjen de Vries appointed as full professor of Information Retrieval

Arjen P. de Vries appointed as full professor of Information Retrieval at Radboud University.

WWW.RU.NL

Jeremy Pickens, Max Hinne, Martine Zwiers and 22 others like this.

Mounia Lalmas-Roelleke Fantastic news.

Unlike · Reply · 1 · 15 mins

Miriam Gravemaker Gefeliciteerd!

Unlike · Reply · 1 · 9 mins

Mani Zandifar Congratulations!

Unlike · Reply · 1 · 8 mins

Katya Mouriits Congratulations! And wow, big change!

Unlike · Reply · 1 · 7 mins

Raffaele Perego Great! Congratulations Arjen!

Unlike · Reply · 1 · 6 mins
Music

Collaborative Filtering
Collaborative Filtering

- Collaborative filtering (originally introduced by Patti Maes as “social information filtering”)
  1. Compare user judgments
  2. Recommend differences between similar users

- Leading principle: People’s tastes are not randomly distributed
  – A.k.a. “You are what you buy”
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Boris</td>
<td>★★★☆☆☆</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Dave</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Will</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>George</td>
<td>★★★★★☆</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td></td>
<td>Boris</td>
<td>Dave</td>
<td>Will</td>
<td>George</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Die Hard</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mission: Impossible</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoldenEye</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casino Royale</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanic</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notting Hill</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridget Jones’s Diary</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Love Actually</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td>🌟🌟🌟🌟🌟</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Items</td>
<td>Boris</td>
<td>Dave</td>
<td>Will</td>
<td>George</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Die Hard</td>
<td>★★★★☆</td>
<td></td>
<td>★★★★</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mission: Impossible</td>
<td></td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoldenEye</td>
<td>★★★★☆</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casino Royale</td>
<td>★★★★☆</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanic</td>
<td>★★★★☆</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notting Hill</td>
<td>★★★★☆</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridget Jones’s Diary</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Love Actually</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️</td>
<td>☀️☀️★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Movie</td>
<td>Boris</td>
<td>Dave</td>
<td>Will</td>
<td>George</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Die Hard</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mission: Impossible</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoldenEye</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casino Royale</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notting Hill</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridget Jones's Diary</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Love Actually</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## User Profile

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Boris</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Dave</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Will</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>George</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>Item Profile</td>
<td>Die Hard</td>
<td>Mission: Impossible</td>
<td>GoldenEye</td>
<td>Casino Royale</td>
<td>Titanic</td>
<td>Notting Hill</td>
<td>Bridget Jones's Diary</td>
<td>Love Actually</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------------------</td>
<td>------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Boris</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Dave</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Will</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>George</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Movie Title</td>
<td>Boris</td>
<td>Dave</td>
<td>Will</td>
<td>George</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Die Hard</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>4★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mission: Impossible</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoldenEye</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casino Royale</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanic</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notting Hill</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridget Jones's Diary</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Love Actually</td>
<td>4★</td>
<td>5★</td>
<td>5★</td>
<td>5★</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The unknown rating for Love Actually is indicated by a question mark.
Collaborative Filtering

If user Boris watched *Love Actually*, how would he rate it?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Boris</td>
<td>★★★☆☆☆</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★☆☆☆</td>
<td>★★★★★★☆☆☆☆☆☆</td>
<td>?</td>
</tr>
<tr>
<td>Dave</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★★☆☆☆☆☆☆</td>
<td>★★★★★☆☆☆☆☆☆☆☆☆</td>
</tr>
<tr>
<td>Will</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★★☆☆☆☆☆☆</td>
<td>★★★★★☆☆☆☆☆☆☆☆☆</td>
</tr>
<tr>
<td>George</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★★☆☆☆☆☆☆</td>
<td>★★★★★☆☆☆☆☆☆☆☆☆</td>
</tr>
</tbody>
</table>
Collaborative Filtering

- Standard item-based formulation (Adomavicius & Tuzhilin 2005)

\[
\text{rat}(u, i) = \frac{\sum_{j \in I_u} \text{sim}(i, j) \times \text{rat}(u, j)}{\sum_{j \in I_u} \text{sim}(i, j)}
\]
Collaborative Filtering

• Benefits over content-based approach
  – Overcomes problems with finding suitable features to represent e.g. art, music
  – Serendipity
  – Implicit mechanism for qualitative aspects like style

• Problems: large groups, broad domains
Prediction vs. Ranking

• Original formulations focused on modelling the users’ item *ratings*: **rating prediction**
  – Evaluation of algorithms (e.g., Netflix prize) by Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) between predicted and actual ratings
Netflix Never Used Its $1 Million Algorithm Due To Engineering Costs

BY CASEY JOHNSTON, ARS TECHNICA  04.16.12 | 8:20 AM | PERMALINK

Netflix awarded a $1 million prize to a developer team in 2009 for an algorithm that increased the accuracy of the company’s recommendation engine by 10 percent. But it doesn’t use the million-dollar code, and has no plans to implement it in the future, Netflix announced on its blog Friday. The post goes on to explain why: a combination of too much engineering effort for the results, and a shift from movie recommendations to the “next level” of personalization caused by the transition of the business from mailed DVDs to video streaming.

Netflix notes that it does still use two algorithms from the team that won the first Progress Prize for an 8.43 percent improvement to the recommendation engine’s root mean squared error (the full $1 million was awarded for a 10 percent improvement). But the increase in accuracy on the winning improvements “did not seem to justify the engineering effort needed to bring them into a production...
Prediction vs. Ranking

• Original formulations focused on modelling the users’ item ratings: rating prediction
  – Evaluation of algorithms (e.g., Netflix prize) by Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) between predicted and actual ratings

• For the end user, the ranking of recommended items is the essential problem: relevance ranking
  – Evaluation by precision at fixed rank (P@N)
Relevance Ranking

- Core problem of Information Retrieval!
Generative Model

- A statistical model for generating data
  - Probability distribution over samples in a given ‘language’

\[
P(\text{\begin{bmatrix} \bullet, \bullet, \bullet, \bullet \end{bmatrix}} | M) = P(\text{\begin{bmatrix} \bullet \end{bmatrix}} | M) \\
P(\text{\begin{bmatrix} \bullet, \bullet \end{bmatrix}} | M, \text{\begin{bmatrix} \bullet \end{bmatrix}}) \\
P(\text{\begin{bmatrix} \bullet, \bullet, \bullet \end{bmatrix}} | M, \text{\begin{bmatrix} \bullet, \bullet \end{bmatrix}}) \\
P(\text{\begin{bmatrix} \bullet, \bullet, \bullet, \bullet \end{bmatrix}} | M, \text{\begin{bmatrix} \bullet, \bullet, \bullet \end{bmatrix}})
\]
Unigram models etc.

\[ P(\bullet \bullet \bullet \bullet) \]

\[ = P(\bullet) P(\bullet | \bullet) P(\bullet | \bullet \bullet) P(\bullet | \bullet \bullet \bullet \bullet) \]

- Unigram Models

\[ P(\bullet) P(\bullet) P(\bullet) P(\bullet) \]

- N-gram Models (here, N=2)

\[ P(\bullet) P(\bullet | \bullet) P(\bullet | \bullet) P(\bullet | \bullet) \]

© Victor Lavrenko, Aug. 2002
Fundamental Problem

- Usually we don’t know the model $M$
  - But have a sample representative of that model

$$P \left( \bullet \bullet \bullet \bullet \mid M \left( \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \right) \right)$$

- First estimate a model from a sample
- Then compute the observation probability

© Victor Lavrenko, Aug. 2002
Unigram Language Models (LM)
– Urn metaphor

\[ P(\bullet \cdot \bullet \cdot \bullet) \sim P(\bullet) P(\cdot) P(\bullet) P(\bullet) P(\bullet) \]
\[ = \frac{4}{9} \times \frac{2}{9} \times \frac{4}{9} \times \frac{3}{9} \]

© Victor Lavrenko, Aug. 2002
• Rank models (documents) by probability of generating the query:

\[
P\left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) | \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) = \frac{4}{9} \times \frac{2}{9} \times \frac{4}{9} \times \frac{3}{9} = \frac{96}{9}
\]

\[
P\left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) | \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) = \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} = \frac{81}{9}
\]

\[
P\left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) | \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) = \frac{2}{9} \times \frac{3}{9} \times \frac{2}{9} \times \frac{4}{9} = \frac{48}{9}
\]

\[
P\left(\begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) | \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet
\end{array}\right) = \frac{2}{9} \times \frac{5}{9} \times \frac{2}{9} \times \frac{2}{9} = \frac{40}{9}
\]
Zero-frequency Problem

• Suppose some event not in our example
  – Model will assign zero probability to that event
  – And to any set of events involving the unseen event

• Happens frequently in natural language text, and it is incorrect to infer zero probabilities
  – Especially when dealing with incomplete samples
Smoothing

• Idea:
  Shift part of probability mass to unseen events

• Interpolate document-based model with a background model (of “general English”)
  – Reflects expected frequency of events
  – Plays role of IDF

$$\lambda + (1 - \lambda)$$
Relevance Ranking

• **Core problem of Information Retrieval!**
  
  – Question arising naturally:
    Are CF and IR, from a modelling perspective, really two different problems then?

Jun Wang, Arjen P. de Vries, Marcel JT Reinders, A User-Item Relevance Model for Log-Based Collaborative Filtering, ECIR 2006
User-Item Relevance Models

- Idea: CF by a probabilistic retrieval model
User-Item Relevance Models

• Idea: CF by a probabilistic retrieval model
• Treat user profile as query and answer the following question:
  – “What is the probability that this item is relevant to this user, given his or her profile”
• Hereto, apply the language modelling approach to IR as a formal model to compute the user-item relevance
Implicit or explicit relevance?

• Rating-based CF:
  – Users explicitly rate “items”
    
    *We use “items” to represent contents (movie, music, etc.)*

• Log-based CF:
  – User profiles are gathered by logging the interactions. Music play-list, web surf log, etc.
User-Item Relevance Models

• Existing User-based/Item-based approaches
  – Heuristic implementations of “word-of-mouth”
  – Unclear how to best deal with the sparse data!

• User-Item Relevance Models
  – Give probabilistic justification
  – Integrate smoothing to tackle the problem of sparsity
User-Item Relevance Models

Target Item

Other Items that the target user liked

Other users who liked the target item

Target User

User Representation

Item Representation

Relevance
User-Item Relevance Models

• Introduce the following random variables

Users: $U \in \{u_1, \ldots, u_K\}$ Items: $I \in \{i_1, \ldots, i_M\}$

Relevance: $R \in \{r, \bar{r}\}$, $r$ "relevant", $\bar{r}$ "not relevant"

• Rank items by their log odds of relevance

$$R_{SV_U}(I) = \log \frac{P(R = r \mid U, I)}{P(R = \bar{r} \mid U, I)}$$
Item Representation

$im$?

Target Item

$\{ib\}$

Query Items: other Items that the target user liked

Relevance

Target User

Item Representation
User-Item Relevance Models

- **Item representation**
  - Use *items that I liked* to represent target user
  - Assume the item “ratings” are independent
  - Linear interpolation smoothing to address sparsity

\[
P(i_b | i_m, r) = (1 - \lambda)P_{ml}(i_b | i_m, r) + \lambda P_{ml}(i_b | r)
\]

\[
RSV_{u_k}(i_m) = \log \frac{P(r | i_m, u_k)}{P(\bar{r} | i_m, u_k)} = \log \frac{P(u_k | r, i_m)P(r | i_m)}{P(u_k | \bar{r}, i_m)P(\bar{r} | i_m)}
\]

\[= \sum_{i_b : i_b \in L_{u_k} \cap c(i_b, i_m) > 0} \log(1 + \frac{(1 - \lambda)P_{ml}(i_b | i_m, r)}{\lambda P(i_b | r)}) + \log P(i_m | r)
\]

\[\lambda \in [0,1] \text{ is a parameter to adjust the strength of smoothing}\]
User-Item Relevance Models

- Probabilistic justification of Item-based CF
  - The RSV of a target item is the combination of its **popularity** and its **co-occurrence** with items (*query items*) that the target user liked.

\[
RSV_{u_k}(i_m) = \sum_{\forall i_b, i_b \in L_{u_k} \cap c(i_b, i_m) > 0} \log(1 + \frac{(1 - \lambda)P_{ml}(i_b \mid i_m, r)}{\lambda P(i_b \mid r)}) + \log P(i_m \mid r)
\]
User-Item Relevance Models

- Probabilistic justification of Item-based CF
  - The RSV of a target item is the combination of its **popularity** and its **co-occurrence** with items (query items) that the target user liked
    - Item co-occurrence should be **emphasized** if more users express interest in both target & query item
    - Item co-occurrence should be **suppressed** when the popularity of the query item is high

$$RSV_{u_k}(i_m) = \sum_{\forall i_b : i_b \in L_{u_k} \cap c(i_b, i_m) > 0} \log(1 + \frac{(1 - \lambda) P_{ml}(i_b | i_m, r)}{\lambda P(i_b | r)}) + \log P(i_m | r)$$

- Co-occurrence between target item and query item
- Popularity of query item
User Representation

Target Item \( i_m \)\n
\{u_b\} \n
Other users who liked the target item

Target User \( u_k \)

Relevance

\( i_m \)?
User-Item Relevance Models

• User representation
  – Represent target item by users who like it
  – Assume the user profiles are independent
  – Linear interpolation smoothing to address sparsity

\[
P_{ml}(u_b \mid u_k, r) = (1 - \lambda)P_{ml}(u_b \mid u_k, r) + \lambda P_{ml}(u_b \mid r)
\]

\[
\text{RSV}_{u_k}(i_m) = \log \frac{P(r \mid i_m, u_k)}{P(\bar{r} \mid i_m, u_k)} = \log \frac{P(i_m \mid r, u_k)P(r \mid u_k)}{P(i_m \mid \bar{r}, u_k)P(\bar{r} \mid u_k)}
\]

\[
= \sum_{\forall u_b : u_b \in L_m} \log(1 + \frac{(1 - \lambda)P_{ml}(u_b \mid u_k, r)}{\lambda P(u_b \mid r)})
\]

\[
\lambda \in [0,1] \text{ is a parameter to adjust the strength of smoothing}
\]
User-Item Relevance Models

• Probabilistic justification of User-based CF
  – The RSV of a target item towards a target user is calculated by the target user's co-occurrence with other users who liked the target item
    • User co-occurrence is emphasized if more items liked by target user are also liked by the other user
    • User co-occurrence should be suppressed when this user liked many items

\[
\text{RSV}_{u_k}(i_m) = \sum_{\forall u_b: u_b \in L_{im}} \log(1 + \frac{(1-\lambda)P_{ml}(u_b|u_k, r)}{\lambda P(u_b|r)})
\]
Empirical Results

• Data Set:
  – Music play-lists from audioscrobbler.com
  – 428 users and 516 items
  – 80% users as training set and 20% users as test set.
  – Half of items in test set as ground truth, others as user profiles

• Measurement
  – Recommendation Precision:
    \( \frac{\text{num of corrected items}}{\text{num. of recommended}} \)
  – Averaged over 5 runs
  – Compared with the suggestion lib developed in GroupLens
P@N vs. lambda

![Graph showing the relationship between P@N and lambda with different return values.](image-url)
<table>
<thead>
<tr>
<th></th>
<th>Top-1 Item</th>
<th>Top-10 Item</th>
<th>Top-20 Item</th>
<th>Top-40 Item</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UIR-Item</strong></td>
<td>0.62</td>
<td>0.52</td>
<td>0.44</td>
<td>0.35</td>
</tr>
<tr>
<td><strong>Item-TFIDF</strong></td>
<td>0.55</td>
<td>0.47</td>
<td>0.40</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>Item-CosSim</strong></td>
<td>0.56</td>
<td>0.46</td>
<td>0.38</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>Item-CorSim</strong></td>
<td>0.50</td>
<td>0.38</td>
<td>0.33</td>
<td>0.27</td>
</tr>
<tr>
<td><strong>User-CosSim</strong></td>
<td>0.55</td>
<td>0.42</td>
<td>0.34</td>
<td>0.27</td>
</tr>
</tbody>
</table>

(a) Precision

<table>
<thead>
<tr>
<th></th>
<th>Top-1 Item</th>
<th>Top-10 Item</th>
<th>Top-20 Item</th>
<th>Top-40 Item</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UIR-Item</strong></td>
<td>0.02</td>
<td>0.15</td>
<td>0.25</td>
<td>0.40</td>
</tr>
<tr>
<td><strong>Item-TFIDF</strong></td>
<td>0.02</td>
<td>0.15</td>
<td>0.26</td>
<td>0.41</td>
</tr>
<tr>
<td><strong>Item-CosSim</strong></td>
<td>0.02</td>
<td>0.13</td>
<td>0.22</td>
<td>0.35</td>
</tr>
<tr>
<td><strong>Item-CorSim</strong></td>
<td>0.01</td>
<td>0.11</td>
<td>0.19</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>User-CosSim</strong></td>
<td>0.02</td>
<td>0.15</td>
<td>0.25</td>
<td>0.39</td>
</tr>
</tbody>
</table>

(b) Recall
So far...

• User-Item relevance models
  – Give a probabilistic justification for CF
  – Deal with the problem of sparsity
  – Provide state-of-art performance
Rating Prediction?

• Previous log-based CF method predicts nor uses rating information
  – Ranks items solely by usage frequency
  – Appropriate for, e.g., music recommendation in a service like Spotify or personalised TV
... Sorted Item Similarity ...
Sorted User Similarity

\[ u_{a} \]

\[ x_{a,1} \]

\[ x_{a,b} ? \]

\[ x_{a,B} \]
Sparseness

• Whether you choose SIR or SUR, in many cases, the neighborhood extends to include “not-so-similar” users and/or items

• Idea:
  Take into considerations the similar item ratings made by similar users as extra source for prediction

Jun Wang, Arjen P. de Vries, Marcel JT Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, SIGIR 2006
<table>
<thead>
<tr>
<th>Sorted User Similarity</th>
<th>Sorted Item Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_a$</td>
<td>$x_{a,b}$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Rating Prediction**

- SUIR
- SUR
- SIR
- Unknown Rating
Similarity Fusion

\[ I_1 \]
\[ I_2 \]
\[ x_{a,b} \]
\[ x_{k,m} \]

<table>
<thead>
<tr>
<th></th>
<th>( I_1 = 0 )</th>
<th>( I_1 = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_2 = 0 )</td>
<td>( x_{a,b} \in SIR )</td>
<td>( x_{a,b} \in SUR )</td>
</tr>
<tr>
<td>( I_2 = 1 )</td>
<td>( x_{a,b} \in SUIR )</td>
<td>( x_{a,b} \in SUIR )</td>
</tr>
</tbody>
</table>
Sketch of Derivation

\[ P(x_{k,m} \mid SUR, SIR, SUIR) \]

\[ = \sum_{I_2} P(x_{k,m}, I_2 \mid SUR, SIR, SUIR)P(I_2) \]

\[ = P(x_{k,m}, I_2 = 1 \mid SUR, SIR, SUIR)P(I_2 = 1) + P(x_{k,m}, I_2 = 0 \mid SUR, SIR, SUIR)(1 - P(I_2 = 1)) \]

\[ = P(x_{k,m} \mid SUIR)\delta + P(x_{k,m} \mid SUR, SIR)(1 - \delta) \]

\[ = P(x_{k,m} \mid SUIR)\delta + (P(x_{k,m} \mid SUR)\lambda + P(x_{k,m} \mid SIR)(1 - \lambda))(1 - \delta) \]

See SIGIR 2006 paper for more details
User-Item Relevance Models

- Information Retrieval Field
- Machine Learning Field

User Representation
Item Representation

Combination rules

Similarity Fusion
- Individual Predictor

Relevance Feedback, Query expansion etc

Latent Predictor Space, Latent semantic analysis, manifold learning etc.
Remarks

- SIGIR 2006 paper estimates probabilities directly from the similarity distance given between users and items.

- TOIS 2008 paper below applies Parzen window kernel density estimation to the rating data itself, to give a full probabilistic derivation:
  - Shows how the “kernel trick” lets us generalize the distance measure; such that a cosine (projection) kernel (length-normalized dot product) can be chosen, while keeping Gaussian kernel Parzen windows.

Relevance Feedback

• Relevance Models for query expansion in IR
  – Language model estimated from known relevant or from top-k documents (Pseudo-RFB)
  – Expand query with terms generated by the LM
• Application to recommendation
  – User profile used to identify neighbourhood; a Relevance Model estimated from that neighbourhood used to expand the profile
  – Deploy probabilistic clustering method PPC to construct the neighbourhood
  – Very good empirical results on P@N

Follow-up question: Can we go beyond “model level” equivalences observed so far, and actually cast the CF problem such that we can use the full IR machinery?

IR System

Query Process

Input

Term occurrences (term-doc matrix)

Inverted Index

Text Retrieval Engine

Output
CF RecSys?!

User Profile Process

User profile (as query)

User Profiles (User-item matrix) → Item Similarity → Inverted Index → Text Retrieval Engine → Output
• Standard item-based formulation

\[
rat(u,i) = \sum_{j \in I_u} \frac{\sum_{j \in I_u} \text{sim}(i,j) \cdot rat(u,j)}{\sum_{j \in I_u} \text{sim}(i,j)}
\]

• More general

\[
rat(u,i) = \sum_{j \in g(u)} f(u,i,j) = \sum_{j \in g(u)} f_1(u,j) \cdot f_2(i,j)
\]

Table 2. User and item components for function \( f \) in user- and item-based CF. \( E \) represents the space where \( e \) belongs, that is, \( e \in E \).
• In (Metzler & Zaragoza, 2009)

\[ s(q,d) = \sum_{t \in g(q)} s(q,d,t) \]

– In particular: factored form

\[ s(q,d,t) = w_1(q,t)w_2(d,t) \]
• Examples

– TF: \[ w_1(q,t) = qf(t) \]
\[ w_2(d,t) = tf(t,d) \]

– TF-IDF: \[ w_1(q,t) = qf(t) \]
\[ w_2(d,t) = tf(t,d) \log \left( \frac{N}{df(t)} \right) \]

– BM25:
\[ w(q,t)_1 = \frac{(k_3 + 1)qf(t)}{k_3 + qf(t)} \]
\[ w(d,t)_2 = \log \left( \frac{N - df(t) + 0.5}{df(t) + 0.5} \right) \frac{(k_1 + 1)tf(t,d)}{k_1 \left( 1 - b + b \cdot \frac{dl(d)}{dl} \right) + tf(t,d)} \]
In item-based Collaborative Filtering

\[
\text{tf}(t,d) = \text{sim}(i,j) \\
\text{qf}(t) = \text{rat}(u,j)
\]

Apply different models

- With different normalizations and norms: \( s_{qd} \), \( L_1 \) and \( L_2 \)

<table>
<thead>
<tr>
<th>Pair</th>
<th>Document</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_{qd} )</td>
<td>No norm</td>
<td>Norm ( (</td>
</tr>
<tr>
<td>( s_{00} )</td>
<td>No norm</td>
<td>( s_{01} )</td>
</tr>
<tr>
<td>( s_{10} )</td>
<td>Norm ( (</td>
<td>Q</td>
</tr>
</tbody>
</table>
IR =~ CF!

• TF L1 s01 is equivalent to item-based CF

\[
s(q,d) = \sum_{t \in g(q)} w_1(q,t) w_2(d,t) = \sum_{t \in g(q)} qf(t) \frac{tf(t,d)}{\sum_{t \in g(q)} tf(t,d)}
\]

\[
rat(u,i) = \sum_{j \in I_u} rat(u,j) \frac{\text{sim}(i,j)}{\sum_{j \in I_u} \text{sim}(i,j)}
\]

\[
\begin{align*}
tf(t,d) &= \text{sim}(i,j) \\
qf(t) &= \text{rat}(u,j)
\end{align*}
\]
Empirical Results

• Movielens 1M
  – Movielens100k: comparable results

![Graph showing nDCG values for different models and parameters]
Vector Space Model

• Challenge:
  – No shared “words” to relate documents to queries

• Solution:
  – First project users and items in a common space

• Two extreme settings:
  – Project users into a space with dimensionality of the number of items
  – Project items into a space with dimensionality of the number of users

A. Bellogín, J. Wang, P. Castells. *Bridging Memory-Based Collaborative Filtering and Text Retrieval*. Information Retrieval Journal
Item Space

- User

\[ u_I = (r_{u1}, \cdots, r_{uk}, \cdots, r_{un}) \]

- Item

\[ i_I = (s_{i1}, \cdots, s_{ik}, \cdots, s_{in}) \]

- Rank

\[ \text{Score}(u_I, i_I) = \sum_k r_{uk} \cdot s_{ik} \]

- Predict rating:

\[ \hat{r}(u, i) = \frac{\sum_{k=1}^{n} r_{uk} \cdot s_{ik}}{\sum_{\forall k: r_{uk} \neq 0} s_{ik}} = \frac{\text{Score}(u_I, i_I)}{\sum_{\forall k: r_{uk} \neq 0} s_{ik}} = \frac{\text{Score}(u_I, i_I)}{\text{Score}(\delta(u_I), i_I)} \]
User space

- User
  \[ u_U = (s_{u1}, \ldots, s_{uk}, \ldots, s_{um}) \]

- Item
  \[ i_U = (r_{1i}, \ldots, r_{ki}, \ldots, r_{mi}) \]

- Rank
  \[ \text{Score}(u_I, i_I) = \sum_k r_{uk} \cdot s_{ik} \]

- Predict rating:
  \[ \hat{r}(u, i) = \frac{\sum_{k=1}^{m} r_{ki} \cdot s_{uk}}{\sum_{\forall k: r_{ki} \neq 0} s_{uk}} = \frac{\text{Score}(u_U, i_U)}{\sum_{\forall k: r_{ki} \neq 0} s_{uk}} = \frac{\text{Score}(u_U, i_U)}{\text{Score}(u_U, \delta(i_U))} \]
• Users and items in shared orthonormal space:
  \[ u^v = \lambda_1^v e_1 + \cdots + \lambda_l^v e_l = (\lambda_1^v, \cdots, \lambda_l^v) \]
  \[ i^j = \mu_1^j e_1 + \cdots + \mu_l^j e_l = (\mu_1^j, \cdots, \mu_l^j) \]

• Consider covariance matrix
  \[ C_I = \text{cov}(X) \quad a_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)] \]
  \[ \mu_i = E(X_i) \]

• Spectral theorem now states that an orthonormal basis of eigenvectors exists.
Linear Algebra

• Use this basis to represent items and users:

\[ \mathbf{i}^j = \mu_1^j e_1 + \cdots + \mu_n^j e_n = (\mu_1^j, \ldots, \mu_n^j) \]

\[ \mathbf{u}^\mathbf{v} = r_{v1} \mathbf{i}^1 + \cdots + r_{vn} \mathbf{i}^n = r_{v1} \sum_j \mu_j^1 e_j + \cdots + r_{vn} \sum_j \mu_j^n e_j \]

\[ = (r_{v1} \mu_1^1 + \cdots + r_{vn} \mu_1^n, \ldots, r_{v1} \mu_n^1 + \cdots + r_{vn} \mu_n^n) \]

\[ = (r_{v1} + \cdots + r_{vn}) \cdot C_I \]

• The dot product then has a remarkable form (of the IR models discussed):

\[ \mathbf{u}^\mathbf{v} \cdot \mathbf{i}^j = \sum_{p=1}^n \mu_p^j (r_{v1} \mu_p^1 + \cdots + r_{vn} \mu_p^n) = \sum_{p=1}^n \mu_p^j \sum_{k=1}^n r_{vk} \mu_p^k = \sum_{k=1}^n r_{vk} \sum_{p=1}^n \mu_p^j \mu_p^k \]
Subspaces…

• Number of items (n) vs. number of users (m):
  – If \( n < m \), a linear dependency must exist between users in terms of the item space components
  – In this case, it has been known empirically that item-based algorithms tend to perform better
    • Dimension of sub-space key for the performance of the algorithm?
    • ~ better estimation (more data per item) in the probabilistic versions
Subspaces…

• Matrix Factorization methods are captured by assuming a lower-dimensionality space to project items and users into (usually considered “model-based” rather than “memory-based”)

\[ u^v \cdot i^j = \sum_{p=1}^{n} \mu_p^j (r_{v1}\mu_1^p + \cdots + r_{vn}\mu_n^p) = \sum_{p=1}^{l} \mu_p^j \sum_{k=1}^{n} r_{vk}\mu_p^k \]

~ Latent Semantic Indexing (a VSM method replicated as pLSA and variants)
Ratings into Inverted File

- Note: distribution of item occurrences not Zipfian like text, so existing implementations (including choice of compression etc.) may be sub-optimal for CF runtime performance
Weighting schemes under the unified framework for item-based CF. The rating from the (query) user $u$ is denoted as $r_{uk}$, the similarity between the target item and item $k$ is $s_{ik}$, $N$ is the number of items, $N_k$ is the number of items similar to item $k$, $il(i)$ is the number of similar items of the target item, and $\bar{il}$ is the average $il$.

<table>
<thead>
<tr>
<th>Method</th>
<th>$w_{uk}^u$</th>
<th>$w_{ik}^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>1 if rated</td>
<td>1 if similar</td>
</tr>
<tr>
<td>TF</td>
<td>$r_{uk}$</td>
<td>$s_{ik}$</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>$r_{uk}$</td>
<td>$s_{ik} \log \left( \frac{N}{N_k} \right)$</td>
</tr>
<tr>
<td>BM25</td>
<td>$\frac{(k_3+1)r_{uk}}{k_3+r_{uk}}$</td>
<td>$\log \left( \frac{N-N_k}{N_k} \right) \cdot \frac{(k_1+1)s_{ik}}{k_1 \left( (1-b)+b \cdot \frac{il(i)}{\bar{il}} \right) + s_{ik}}$</td>
</tr>
<tr>
<td>Language Model (Jelinek-Mercer)</td>
<td>$r_{uk}$</td>
<td>$(1-\lambda)p(k</td>
</tr>
<tr>
<td>Language Model (Dirichlet)</td>
<td>$r_{uk}$</td>
<td>$\frac{s_{ik}}{il(i)+\mu} + \mu \frac{p(k</td>
</tr>
</tbody>
</table>
### Table 8: Performance results in the user space for the item ranking task (MovieLens 1M).

<table>
<thead>
<tr>
<th>Methods</th>
<th>P@5</th>
<th>P@10</th>
<th>nDCG@3</th>
<th>nDCG@5</th>
<th>nDCG@10</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-based CF</td>
<td>0.0274</td>
<td>0.0252</td>
<td>0.0224</td>
<td>0.0232</td>
<td>0.0224</td>
<td>0.0139</td>
</tr>
<tr>
<td>TF $</td>
<td></td>
<td>n_{10}</td>
<td></td>
<td>_1$</td>
<td>0.0274</td>
<td>0.0252</td>
</tr>
<tr>
<td>MF</td>
<td>0.0623</td>
<td>0.0586</td>
<td>0.0592</td>
<td>0.0606</td>
<td>0.0602</td>
<td>0.0307</td>
</tr>
<tr>
<td>BM25 $</td>
<td></td>
<td>n_{01}</td>
<td></td>
<td>_2$</td>
<td>0.0983</td>
<td>0.0863</td>
</tr>
<tr>
<td>TF-IDF $</td>
<td></td>
<td>n_{01}</td>
<td></td>
<td>_2$</td>
<td>0.0984</td>
<td>0.0862</td>
</tr>
<tr>
<td>Dirichlet $n_{00}$</td>
<td>0.1013</td>
<td>0.0892</td>
<td>0.1020</td>
<td>0.0953</td>
<td>0.0921</td>
<td>0.0395</td>
</tr>
<tr>
<td>BM25 $n_{00}$</td>
<td>0.1013</td>
<td>0.0892</td>
<td>0.1020</td>
<td>0.0953</td>
<td>0.0921</td>
<td>0.0395</td>
</tr>
<tr>
<td>Jelinek-Mercer $n_{00}$</td>
<td>0.1013</td>
<td>0.0892</td>
<td>0.1020</td>
<td>0.0953</td>
<td>0.0921</td>
<td>0.0395</td>
</tr>
<tr>
<td>TF-IDF $n_{00}$</td>
<td>0.1013</td>
<td>0.0892</td>
<td>0.1020</td>
<td>0.0953</td>
<td>0.0921</td>
<td>0.0395</td>
</tr>
<tr>
<td>BM25 $</td>
<td></td>
<td>n_{10}</td>
<td></td>
<td>_2$</td>
<td>0.1038</td>
<td>0.0902</td>
</tr>
<tr>
<td>TF-IDF $</td>
<td></td>
<td>n_{10}</td>
<td></td>
<td>_2$</td>
<td>0.1041</td>
<td>0.0902</td>
</tr>
</tbody>
</table>
Table 10 Performance results in the user space for the item ranking task (MovieLens 10M).

<table>
<thead>
<tr>
<th>Methods</th>
<th>P@5</th>
<th>R@5</th>
<th>nDCG@5</th>
<th>MAP</th>
<th>MRR</th>
<th>bpref</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-based CF</td>
<td>0.0124</td>
<td>0.0018</td>
<td>0.0102</td>
<td>0.0090</td>
<td>0.0425</td>
<td>0.4972</td>
</tr>
<tr>
<td>TF $|n_{10}|_1$</td>
<td>0.0124</td>
<td>0.0018</td>
<td>0.0102</td>
<td>0.0090</td>
<td>0.0425</td>
<td>0.4972</td>
</tr>
<tr>
<td>MF</td>
<td>0.0456</td>
<td>0.0103</td>
<td>0.0467</td>
<td>0.0162</td>
<td>0.1210</td>
<td>0.3303</td>
</tr>
<tr>
<td>BM25 $|n_{01}|_2$</td>
<td>0.0865</td>
<td>0.0272</td>
<td>0.0773</td>
<td>0.0381</td>
<td>0.2177</td>
<td>0.5983</td>
</tr>
<tr>
<td>TF-IDF $|n_{01}|_2$</td>
<td>0.0865</td>
<td>0.0272</td>
<td>0.0773</td>
<td>0.0381</td>
<td>0.2177</td>
<td>0.5983</td>
</tr>
<tr>
<td>Dirichlet $n_{00}$</td>
<td>0.0913</td>
<td>0.0279</td>
<td>0.0826</td>
<td>0.0388</td>
<td>0.2251</td>
<td>0.5800</td>
</tr>
<tr>
<td>BM25 $n_{00}$</td>
<td>0.0913</td>
<td>0.0279</td>
<td>0.0826</td>
<td>0.0388</td>
<td>0.2251</td>
<td>0.5800</td>
</tr>
<tr>
<td>Jelinek-Mercer $n_{00}$</td>
<td>0.0913</td>
<td>0.0279</td>
<td>0.0826</td>
<td>0.0388</td>
<td>0.2251</td>
<td>0.5800</td>
</tr>
<tr>
<td>TF-IDF $n_{00}$</td>
<td>0.0913</td>
<td>0.0279</td>
<td>0.0826</td>
<td>0.0388</td>
<td>0.2251</td>
<td>0.5800</td>
</tr>
<tr>
<td>TF-IDF $|n_{10}|_2$</td>
<td>0.0927</td>
<td>0.0275</td>
<td>0.0848</td>
<td>0.0382</td>
<td>0.2281</td>
<td>0.5705</td>
</tr>
<tr>
<td>BM25 $|n_{10}|_2$</td>
<td>0.0928</td>
<td>0.0277</td>
<td>0.0850</td>
<td>0.0382</td>
<td>0.2285</td>
<td>0.5716</td>
</tr>
</tbody>
</table>
Table 12 Results for the rating prediction task (*Movielens 1M*).

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item-based CF</td>
<td>0.8210&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.0255&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>MF</td>
<td>0.6747&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.8687&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>BM25</td>
<td>0.8236&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.0408&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>0.8256&lt;sup&gt;c&lt;/sup&gt;</td>
<td>1.0301&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>0.8284&lt;sup&gt;d&lt;/sup&gt;</td>
<td>1.0359&lt;sup&gt;d&lt;/sup&gt;</td>
</tr>
<tr>
<td>Jelinek-Mercer</td>
<td>0.8290&lt;sup&gt;d&lt;/sup&gt;</td>
<td>1.0358&lt;sup&gt;d&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-based CF</td>
<td>0.9443&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.2138&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>MF</td>
<td>0.6747&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0.8687&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>BM25</td>
<td>0.9443&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.2138&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>0.9443&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.2138&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>0.9443&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.2138&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Jelinek-Mercer</td>
<td>0.9443&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.2138&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>
Concluding Remarks

• The probabilistic models are elegant (often deploying impressive maths), but what do they really add in understanding IR & CF – i.e., beyond the (often claimed to be “ad-hoc”) approaches of the VSM?
Concluding Remarks

• Clearly, the models in CF & IR are closely related
• Should these then really be studied in two different (albeit overlapping) communities, RecSys vs. SIGIR?
Meanwhile at TREC…
Contextual Suggestions

• Given a user profile and a context, make suggestions
  – AKA Context-aware Recommendation, zero-query Information Retrieval, …
“Entertain me”

• Recommend “things to do”, where
  – User profile consists of opinions about attractions
  – Context consists of a specific geo-location
Given a user profile

- 70 – 100 POIs represented by a title, description and URL (situated in Chicago / Santa Fe)
- Rated on a scale 0 – 4

125, Adler Planetarium & Astronomy Museum, "Interactive exhibits & high-tech sky shows entertain stargazers -- lakefront views are a bonus.", http://www.adlerplanetarium.org/

131, Lincoln Park Zoo, "Lincoln Park Zoo is a free 35-acre zoo located in Lincoln Park in Chicago, Illinois. The zoo was founded in 1868, making it one of the oldest zoos in the U.S. It is also one of a few free admission zoos in the United States.", http://www.lpzoo.org/
• … and a context
  – Corresponding to a metropolitan area in the USA, e.g., 109, Kalamazoo, MI
Suggest Web pages / snippets
– From the Open Web, or from ClueWeb

700, 109, 1,"About KIA History Kalamazoo Institute of Arts  KIA History","The Kalamazoo Institute of Arts is a nonprofit art museum and school. Since, the institute has offered art classes and free admission programming, including exhibitions, lectures, events, activities and a permanent collection. The KIAs mission is to cultivate the creation and appreciation of the visual arts for the communities",clueweb12-1811wb-14-09165
Common approach

\[ P_{rel}(u, s) = P(s) \cdot (\lambda \cdot \text{SIM}(u^+, s) - (1 - \lambda) \cdot \text{SIM}(u^-, s)) \]

- Candidate Selection Prior
- Personalization
• A. Bellogín, J. Wang, P. Castells. Bridging Memory-Based Collaborative Filtering and Text Retrieval. Information Retrieval (to appear)
• Jun Wang, Arjen P. de Vries, Marcel JT Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, SIGIR 2006
• Jun Wang, Arjen P. de Vries, Marcel JT Reinders, A User-Item Relevance Model for Log-Based Collaborative Filtering, ECIR 2006
Thanks

- Alejandro Bellogín
- Jun Wang
- Thijs Westerveld
- Victor Lavrenko