A Recommender System Based on Omni-Channel Customer Data

Matthias Carnein, Leschek Homann, Heike Trautmann, Gottfried Vossen

University of Münster, Germany
17th July 2019
Recommender Systems

- Recommender systems suggest products to users that might be of interest to them
- They help customers sift through the product catalogue and discover new products

Customers who bought this item also bought

- AmazonBasics Mid-Back Office Chair, Black
  - #1 Best Seller in Managerial Chairs & Executive
  - $64.99 with Prime

- Furmax Office Chair PU Leather Gaming Chair, High Back Ergonomic Racing Chair, Desk Chair...
  - #1 Best Seller in Office Drafting Chairs
  - $66

- BestOffice High-back Computer Racking Gaming Chair
  - 5 Stars
  - $85.99

- Ergonomic Mesh Computer Office Desk Midback Task Chair w/Metal Base, One Pack
  - 5 Stars
  - $46.99

- Devoko Mid-Back Computer Office Swivel Desk Chair With Arms Height Adjustable...
Case Study

- Spanish retailer in the home décor business
- > 100 stores and an online shop
- In total 2.2 million transactions and 800,000 members of a loyalty programme

Female: 90%
Male: 10%

Age distribution:
- 0-20: 100,000
- 20-40: 80,000
- 40-60: 50,000
- 60-80: 20,000
- 80-100: 0
A Recommender System Based on Omni-Channel Customer Data

User-Item Matrix

▶ **State-of-the-art**: analyse what ‘like-minded’ users enjoy
▶ Often based on explicit feedback (e.g. movie ratings)
▶ In retailing, interest needs to be inferred from implicit feedback, e.g. purchase history
▶ Preferences are stored in a large but sparse **User-Item matrix**

```
<table>
<thead>
<tr>
<th></th>
<th>i_1</th>
<th>i_2</th>
<th>i_3</th>
<th>i_4</th>
<th>i_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>u_2</td>
<td>1</td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>u_3</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u_4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
```
1. Omni-Channel
Omni-Channel

- Using the **purchase history works well** as an indication of interest
- But there is much more information available to infer preference
- **Online Search** and **Surf Behaviour, Newsletters** and **Social Media**
Website – Searches

For confidentiality reasons, the example shows an unrelated company.
Weblogs – Searches

- Use the **search behaviour on the website** as an indicator for interest
- Analyse the stored HTML-GET requests for search queries (‘q=’)
- Match the query to product name, description or product family
- Use cookies to identify users after they logged in once

**USER SEARCH:** red towel  
**SERVER LOG:** GET/ www.url.com/q=red towel
Website – Visits

* for confidentiality reasons, the example shows an unrelated company
Weblogs – Visits

- Use the **website visits** as an indication of interest
- Analyse the stored HTML-GET requests for product pages
- Use cookies to identify users after they logged in once

**VISITED CATEGORY:** bags  
**SERVER LOG:** GET/ www.url.com/bags.html
Company sends out newsletters for new products and discounts

Record **newsletter opens** and **click throughs** as an indication of preference

Identify users based on personalised link / cookie

* for confidentiality reasons, the example shows an unrelated company
Social Media

- Company posts promotions to facebook
- Use facebook comments and likes as an indication of interest
- Identify users based on name

* for confidentiality reasons, the example shows an unrelated company
A Recommender System Based on Omni-Channel Customer Data

Omni-Channel

Purchase History: purchases
Weblog: search $s$ and visit $v$
Social Media: post $p$ and like $l$
Newsletter: open $o$ and click $c$

Preference Matrix

(weighted) sum
Information per Channel

- Purchases: 643,409
- Website Searches: 48,127
- Website Visits: 2,410,111
- Newsletter Open: 544,075
- Newsletter Click: 28,359
- Facebook: 396,662

Item Ratings

Matthias Carnein
Store Location

- Detect WiFi probe requests from phones
- Identify users based on purchases with MAC address present
- **Possible Scenario**: Trigger a recommendation when customer visited a store but left without a purchase
- Because of **privacy concerns**, we did not make use of this data
2. **Recommendations**
k-Nearest Neighbour ($k$NN)

- Weighted average of ‘like-minded’ users [Liu 2011]
- Improvement: Weighted average over items

**User-based CF**

<table>
<thead>
<tr>
<th></th>
<th>$i_1$</th>
<th>$i_2$</th>
<th>$i_3$</th>
<th>$i_4$</th>
<th>$i_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td>1</td>
<td>?</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_4$</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

$\hat{r}_{2,3} = 3$

**Item-based CF**

<table>
<thead>
<tr>
<th></th>
<th>$i_1$</th>
<th>$i_2$</th>
<th>$i_3$</th>
<th>$i_4$</th>
<th>$i_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td>1</td>
<td>?</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_4$</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

$\hat{r}_{2,3} = 3$
Singular Value Decomposition (SVD)

- Split the large User-Item matrix in two smaller matrices [Funk 2018]
- By detecting hidden characteristics that approximate the purchase behaviour
- Special case for implicit feedback using Alternating Least Squares [Hu et al. 2008]
Co-Clustering

- Cluster users and items such that prediction error is minimized [George et al. 2005]
- Implicitly forms co-clusters of users and items
- Use the average rating of user, item, and User-Item cluster for prediction
 Baseline

- For comparison we include a baseline algorithm
- Randomly recommends by sampling from the User-Item matrix

\[
\begin{array}{ccccc}
  & i_1 & i_2 & i_3 & i_4 & i_5 \\
 u_1 & 3 & 0 & 4 & 0 & 4 \\
 u_2 & 1 & 0 & ? & 2 & 3 \\
 u_3 & 0 & 5 & 0 & 0 & 0 \\
 u_4 & 1 & 0 & 3 & 2 & 3 \\
\end{array}
\]

\[\hat{r}_{2,3} = \text{random}(r_{u,i})\]
3. Evaluation
Masking

User-Item Matrix

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Fill Zero & Binarize

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Train set

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Test set

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Matthias Carnein
Omni-Channel Recommendations
Evaluation
Conclusion
Rank Measure

- Ideal: Relevant products should be ranked higher
- Measure: Average percentile-rank of items where the user has interest

<table>
<thead>
<tr>
<th>Product</th>
<th>Percentile Rank</th>
<th>Purchase</th>
<th>Rank Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandellier</td>
<td>0%</td>
<td>✔️</td>
<td>0%</td>
</tr>
<tr>
<td>Blackout Curtain</td>
<td>3%</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Wall Clock</td>
<td>6%</td>
<td>✔️</td>
<td>6%</td>
</tr>
<tr>
<td>Towel</td>
<td>97%</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Canvas Art</td>
<td>100%</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>
Recommendation Quality

![Chart showing recommendation quality for different methods: SVD (ALS), SVD, Co-Clustering, Item-based CF, and Baseline. The chart indicates that Baseline has the highest rank of 49.99, followed by Item-based CF with 11.11, Co-Clustering with 16.97, SVD with 34.10, and SVD (ALS) with 7.76.]
4. Conclusion
Conclusion

- Recommender System based on **Omni-Channel Customer Data**
- Using information from purchases, search and browsing behaviour as well as newsletter and Facebook interactions
- Implemented using a real-life case
- Very high recommendation accuracy

Outlook:

- Improve customer recognition across the different channels
- Observe recommendation performance in real-life
References


