Using Navigation to Improve Recommendations in Real-Time

Chao-Yuan Wu\textsuperscript{1}, Chris Alvino\textsuperscript{2}, Alex Smola\textsuperscript{3}, Justin Basilico\textsuperscript{2}
What contents to show?
Traditional recommender systems

History data → Future behavior
Watching alone, with friends, or with children?
Who is watching? (shared accounts)
May depend on your mood, how much time you have, etc.
Challenges

1. User intention depends on **unpredictable factors**
2. User intention can **vary significantly between sessions**
Current solutions

1. Use context information
   a. Time of day, day or week, device etc.

2. Present a mixture
Current solutions

1. Use context information
   a. Time of day, day or week, device etc.
2. Present a mixture

Can we do better?
Navigation gives valuable information about user intent.
Observe navigation signals
Update unseen rows in real-time

Navigation signal

Scroll

Skip
Online adaptation framework
Completely general - whenever contents don’t fit on single display
Focus on a case based on Netflix PS3 UI

1. Contents organized in rows
2. Horizontal or vertical scrolls
Focus on a case based on Netflix PS3 UI

1. Contents organized in rows
2. Horizontal or vertical scrolls
3. Scrolls as navigation signals
4. Consider **reordering rows**, while videos in a row are fixed
Model & Algorithm
# Videos organized in rows

<table>
<thead>
<tr>
<th>Action Movies</th>
<th>Comedies</th>
<th>Documentaries</th>
<th>Sci-Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Play Button" /></td>
<td><img src="image2.png" alt="Play Button" /></td>
<td><img src="image3.png" alt="Play Button" /></td>
<td><img src="image4.png" alt="Play Button" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Play Button" /></td>
<td><img src="image6.png" alt="Play Button" /></td>
<td><img src="image7.png" alt="Play Button" /></td>
<td><img src="image8.png" alt="Play Button" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Play Button" /></td>
<td><img src="image10.png" alt="Play Button" /></td>
<td><img src="image11.png" alt="Play Button" /></td>
<td><img src="image12.png" alt="Play Button" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Play Button" /></td>
<td><img src="image14.png" alt="Play Button" /></td>
<td><img src="image15.png" alt="Play Button" /></td>
<td><img src="image16.png" alt="Play Button" /></td>
</tr>
<tr>
<td><img src="image17.png" alt="Play Button" /></td>
<td><img src="image18.png" alt="Play Button" /></td>
<td><img src="image19.png" alt="Play Button" /></td>
<td><img src="image20.png" alt="Play Button" /></td>
</tr>
<tr>
<td><img src="image21.png" alt="Play Button" /></td>
<td><img src="image22.png" alt="Play Button" /></td>
<td><img src="image23.png" alt="Play Button" /></td>
<td><img src="image24.png" alt="Play Button" /></td>
</tr>
<tr>
<td>Latent “interest” variable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Movies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comedies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentaries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sci-Fi</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Only scroll or play from rows that user is interested in
<table>
<thead>
<tr>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Movies</td>
</tr>
<tr>
<td>Comedies</td>
</tr>
<tr>
<td>Romantic comedies</td>
</tr>
<tr>
<td>Sci-Fi</td>
</tr>
</tbody>
</table>
User interacts with the page

- Action Movies
- Comedies
- Romantic comedies
- Sci-Fi
### Update intent estimation

<table>
<thead>
<tr>
<th>Action Movies</th>
<th>Comedies</th>
<th>Romantic comedies</th>
<th>Sci-Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>🗙️❌</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td></td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>❌</td>
<td></td>
</tr>
</tbody>
</table>

**Skip**
<table>
<thead>
<tr>
<th>Category</th>
<th>Action Movies</th>
<th>Comedies</th>
<th>Romantic comedies</th>
<th>Sci-Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rearrange page</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skip</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Model

\[ \Theta_0 \rightarrow \Theta_p \rightarrow \delta_p \rightarrow I_{s,r} \rightarrow S_{s,r} \rightarrow C_{s,r,i} \rightarrow \psi_0 \rightarrow \psi_u \rightarrow \Theta_u \rightarrow \Theta_s \text{ (session)} \rightarrow \text{user} \rightarrow \text{row} \rightarrow \text{video} \]
Model

- $\Theta_0$
- $\Theta_\rho$
- $\delta_\rho$
- $I_{s,r}$
- $S_{s,r}$
- $C_{s,r,i}$
- $\psi_\rho$
- $\theta_u$
- $\psi_u$

Scroll

Interest

Play
Online adaptation

1. Observe

2. Update session-specific parameters

3. Update marginal probability
Offline training to estimate long-term preference by EM Algorithm

1. E-step: estimate “interest”
2. M-step: optimize all other variables
Choice of the model

1. Independent of the function class
   a. Inner product, FM, deep networks, etc.
   b. We use submodular functions
      (see e.g. Ahmed et al. WSDM ‘12)

2. Tightly coupled with the UI design
Roundtrips to a server add latency
Efficient in-browser updates with prefetching
Experiments
Dataset

- Netflix PlayStation 3 sessions.
- Training: April-May 2015 (294k sessions)
- Testing: June 2015 (59k sessions)
- 40 rows per page
- 75 videos per row (max)
Conditional play probability increases as the user delves into the list.
Evaluating online adaptation

- Fix 10 rows, and update rows below
- Row considered **positive** if any video in the row is played
- Metrics:
  - Mean reciprocal rank (MRR)
  - Precision @ 5
- Baseline: Factorization Machine over user and row
  - libFM [S. Rendle 2012] implementation
Online adaptation improves recommendations

---

**Gain in MRR (%)**

- **Online update**
- **FM**

**Number of rows observed**

---

**Gain in precision @ 5 (%)**

- **Online update**
- **FM**

**Number of rows observed**
Online adaptation helps with cold-start
Model fatigue effects and repeated plays to get accurate predictions.
Model fatigue effects and repeated plays to get accurate predictions.
Summary

1. Using navigation to
   a. improve recommendations in real-time
   b. help with cold-start
2. **Efficient** in-browser update by pre-fetching
3. Model **fatigue effects** and **repeated plays** to get accurate play predictions
4. **General** framework
   a. Can be applied to many user interfaces