When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation

Malte Ludewig
malte.ludewig@tu-dortmund.de
Joint work with Dietmar Jannach
Session-Based Recommendation

- Continuously adopt to the most recent implicit feedback
  - For example: Item clicks in a user’s shopping session

- Signals can be extracted from past sessions
- Special restriction: No user histories
Session-Based Recommendation

- Continuously adopt to the most recent implicit feedback
  - For example: Item clicks in a user’s shopping session

- Signals can be extracted from past sessions
- Special restriction: No user histories
Session-Based Recommendation

- Continuously adopt to the most recent implicit feedback
  - For example: Item clicks in a user’s shopping session

- Signals can be extracted from past sessions
- Special restriction: No user histories
Session-Based Recommendation

- Continuously adopt to the most recent implicit feedback
  - For example: Item clicks in a user’s shopping session

- Signals can be extracted from past sessions
- Special restriction: No user histories
Recent approach to the problem by Hidasi et al.

Using recurrent neural networks to model sequences in sessions

SGD with pairwise ranking loss functions (BPR, TOP1)

Significantly outperformed an Item-KNN approach
General Motivation

- True advantage of applying new and complex machine learning approaches to recommendation problems not easy to judge
  - Baselines might not be strong enough
  - Dependent on domain, dataset and evaluation method
  - Potential biases
  - Scalability

- Our goal
  - Provide a better understanding for session-based recommendation
  - Propose a simple neighborhood-based baseline for the scenario
  - Comparison with GRU4REC for multiple datasets
Session-Based KNN (S-KNN)

- Given the current session
  - Find $k$ most similar past sessions
  - Cosine similarity of bit vectors
  - Score items by the sum of session similarities

- Getting the similarities for all sessions is slow
  - Only sessions with one item from the current session at minimum
  - Only the $n$ most recent sessions
Datasets for Evaluation

E-Commerce

IJCAI-15 Competition (Tmall)
- 650k sessions over 1 year
- 300k items

RecSys Challenge 15 (RSC15)
- 8M sessions over 6 month
- 37k items

Music

last.fm listening logs
- 120k sessions in 1 month
- 200k items

8tracks.com playlists
- 82k sessions
- 54k items

artofthemix.org playlists
- 82k sessions
- 54k items
Protocol

- Evaluation protocol
  - Sliding window
  - Time-based splitting
  - Recommend for second to last click per session in the test set
  - Measure Hit rate and MRR at list length 20

- Additionally measure...
  - Popularity and catalog coverage
  - Runtimes and memory consumption

- Optimized parameters for each data set (validation set)
Mixed results for a single split on RSC15

S-KNN significantly better in a sliding window evaluation

Bigger difference for Tmall
  - Maybe less sequential patterns in the data
Mixed results for a single split on RSC15

- S-KNN significantly better in a sliding window evaluation
- Bigger difference for Tmall
  - Maybe less sequential patterns in the data
Mixed results for a single split on RSC15

S-KNN significantly better in a sliding window evaluation

Bigger difference for Tmall
  - Maybe less sequential patterns in the data
Using the approaches in a **weighted hybrid** (WH)

- Combining the signals improves both HR@20 and MRR@20
- Different optimal weights for the best HR and MRR
- Training the models with **data from the last $n$ days**
  - Both approaches quite stable regarding the HR@20
  - Focusing the last few days is sufficient for S-KNN
**Additional Measurements**

- **Runtimes and memory consumption**
  - Desktop PC with an Intel i7-4790k on RSC15
  
<table>
<thead>
<tr>
<th></th>
<th>S-KNN</th>
<th>GRU</th>
<th>GRU(GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>90s</td>
<td>23h</td>
<td>8h</td>
</tr>
<tr>
<td>Recommendation</td>
<td>26ms</td>
<td>12ms</td>
<td>12ms</td>
</tr>
<tr>
<td>Memory used</td>
<td>6GB</td>
<td>600MB</td>
<td>600MB</td>
</tr>
</tbody>
</table>

- **Popularity bias**
  - S-KNN recommends more popular items (0.036 vs. 0.028)

- **Catalog coverage**
  - GRU4REC has a slightly higher coverage (47% vs. 41%)

- **Mixed results for music datasets**
  - S-KNN performs better for 8tracks.com and artofthemix.org
  - Advantages for GRU on last.fm
Conclusions & Future Work

- S-KNN shows competitive results
  - Potentially relevant sequential information missed by S-KNN
- Combinations of both approaches show promising results
- Further improvements for RNN-based approaches to be expected
- Meanwhile progress was made
  - GRU4REC
    - **0.636** HR@20 / **0.268** MRR@20
  - Simple heuristic with sequential patterns (see RecTemp `17)
    - **0.690** HR@20 / **0.307** MRR@20
  - New extensions to S-KNN
    - **0.709** HR@20 / **0.304** MRR@20
  - GRU4REC v2 already improved the performance
    - **0.711** HR@20 / **0.310** MRR@20
- Future: Extensive comparison of available methods
  - Which method works best for which data, and why?
When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation

Malte Ludewig
malte.ludewig@tu-dortmund.de

- Questions?