Local Time-Aware Models for Next-Item Recommendation

Maria Kalantzi
Outline

● Motivation
● Problem statement
● The proposed method
● Experimental evaluation
● Conclusions
Local Item-Item Models for Top-N Recommendation

Local item-item models improve upon global item-item model.

Global item-item model and local item-item models yield the same results.

Image Source: https://www-users.cs.umn.edu/~chri2951/
Next-Item Recommendation

Image Source:
https://www.semanticscholar.org/paper/Fusing-Similarity-Models-with-Markov-Chains-for-He-McAuley/fe47e2bcbe852bcb33c90a4d57664e954f0e82a
Outline of the Proposed Method

1. Clustering of the users
2. Global MC model
3. Local MC models
4. Refinement of the users’ assignment to clusters
User Clustering

- **Goal**: create subsets of users with similar tastes and similar sequential behavior.
- We represent each user as a transition matrix.
- The feature space is the set of different transitions across all users.
- Initial clustering by CLUTO or random.
Global-local Markov Chain Models

- Estimate one global model for the whole set of users.
- Estimate one local model for each subset of users that the method discovers.
- A hyperparameter $g$ controls the contribution of the global and local components of the model:

$$\text{Rec}_u = g \times \text{GMC} + (1-g) \times \text{LMC}(u)$$

- Update of the user assignment.
- Different variants of the Markov-Chain Model explored as the base model.
Global-local Markov Chain Models

- The models are jointly optimized with the user assignments.
- Quadratic loss function for updating the user assignment to clusters.
Experimental Evaluation
Experimental Setting

Datasets:

- FourSQ
- Amazon Video Games
- Amazon Movies and TV
- Steam
- Movielens 1M

Baselines:

- BPR-MF
- FPMC
- Fossil
- TransRec
- Caser
- MC1

**Training data**: from each user sequence, we keep everything but the last item.

**Test data**: the last consumed item from each user.

**Evaluation metric**: Hit Rate
## Datasets Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th># users</th>
<th># items</th>
<th># actions</th>
<th>Density(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FourSQ</td>
<td>20,658</td>
<td>9,522</td>
<td>123,741</td>
<td>0.06</td>
</tr>
<tr>
<td>Amazon Video Games</td>
<td>12,379</td>
<td>19,039</td>
<td>160,532</td>
<td>0.07</td>
</tr>
<tr>
<td>Amazon Movies</td>
<td>58,702</td>
<td>72,459</td>
<td>1,216,151</td>
<td>0.03</td>
</tr>
<tr>
<td>Steam</td>
<td>292,004</td>
<td>13,027</td>
<td>3,489,638</td>
<td>0.09</td>
</tr>
<tr>
<td>ML1M</td>
<td>6,040</td>
<td>3,416</td>
<td>999,611</td>
<td>4.84</td>
</tr>
</tbody>
</table>

No repeated items in each user sequence.
Competing Approaches

- **BPR-MF**: It optimizes the matrix factorization with implicit feedback using a pairwise ranking loss. Non-sequential.
- **FPMC**: It captures users’ general taste as well as their sequential behaviors by combining MF with first-order MCs.
- **Fossil**: It combines an item-based similarity method and a higher-order MC model.
- **TransRec**: It embeds items into a transition space where users are modeled as translation vectors operating on item sequences.
- **Caser**: It employs CNN in both horizontal and vertical way to model high-order MCs for sequential recommendation.
- **MC1**: 1st order Markov Chain
Comparison of Markov Chain Models

<table>
<thead>
<tr>
<th></th>
<th>FourSQ</th>
<th>Amazon Video Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMC1</td>
<td>0.2364</td>
<td>0.0643</td>
</tr>
<tr>
<td>LMC1</td>
<td>0.1434</td>
<td>0.0388</td>
</tr>
<tr>
<td>GLMC1</td>
<td>0.2442</td>
<td>0.0715</td>
</tr>
<tr>
<td>GMCcoW20</td>
<td>0.2709</td>
<td>0.0860</td>
</tr>
<tr>
<td>LMCcoW20</td>
<td>0.2091</td>
<td>0.0597</td>
</tr>
<tr>
<td>GLMCcoW20</td>
<td>0.2784</td>
<td>0.0907</td>
</tr>
</tbody>
</table>
## Results (2/2)

Comparison with Competing Approaches

<table>
<thead>
<tr>
<th>HitRate@10</th>
<th>FourSQ</th>
<th>Amazon Video Games</th>
<th>Amazon Movies</th>
<th>ML1M</th>
<th>Steam</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPR-MF</td>
<td>0.2395</td>
<td>0.0277</td>
<td>0.0126</td>
<td>0.0381</td>
<td>0.0550</td>
</tr>
<tr>
<td>FPMC</td>
<td>0.2371</td>
<td>0.0616</td>
<td>0.0237</td>
<td>0.0753</td>
<td>0.0462</td>
</tr>
<tr>
<td>Fossil</td>
<td><strong>0.2822</strong></td>
<td>0.0690</td>
<td>0.0314</td>
<td>0.0570</td>
<td>0.0668</td>
</tr>
<tr>
<td>TransRec</td>
<td>0.2655</td>
<td>0.0617</td>
<td>0.0199</td>
<td>0.0700</td>
<td>0.0645</td>
</tr>
<tr>
<td>Caser</td>
<td>0.2756</td>
<td><strong>0.0944</strong></td>
<td>---</td>
<td><strong>0.2480</strong></td>
<td>---</td>
</tr>
<tr>
<td>MC1</td>
<td>0.2364</td>
<td>0.0643</td>
<td><strong>0.0728</strong></td>
<td>0.2119</td>
<td><strong>0.0736</strong></td>
</tr>
<tr>
<td>GLMC (our method)</td>
<td><strong>0.2791</strong></td>
<td><strong>0.0918</strong></td>
<td><strong>0.0763</strong></td>
<td><strong>0.2495</strong></td>
<td><strong>0.0778</strong></td>
</tr>
</tbody>
</table>
Conclusions

- **Local models** do **improve** upon one globally estimated model for the next-item top-N recommendation.
- The differences in user preferences are important.
- **Global/Local** Probabilistic Models can be a **fast** and efficient **alternative** to complex and time consuming Neural Networks and other sophisticated baselines.
- The refinement of the user assignment to clusters helps and it is data dependent.
- The improvement coming from the refinement of the users’ assignment is not significant potentially because of the data sparsity.
Thank you!